Expert Systems With Applications 174 (2021) 114709

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Novel automatic group identification approaches for group recommendation

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

ARTICLE INFO

Keywords:
Group recommender systems
Bisecting *k*-means clustering
Automatic group identification
User-profiling

ABSTRACT

Group recommender systems are specialized in suggesting preferable products or services to a group of users rather than an individual by aggregating personal preferences of group members. In such expert systems, the initial task is to identify groups of similar users via clustering approaches as user groups are usually not predefined. However, clustering users into groups commonly suffer from sparsity, scalability, and complexity problems as the content in the domain proliferate. Moreover, group homogeneity and size are the critical parameters for organizing group members and enhancing their satisfaction. In this study, we propose novel automatic user grouping approaches by constructing a binary decision tree via bisecting *k*-means clustering for enhanced group formation and group size restriction. Furthermore, we propose adopting a genre-based mapping of user ratings into a tiny and dense vector to represent users, which both improves computation time for constructing the binary decision tree and enables eliminating adverse effects of sparsity. Finally, since the quality of group formation is not only dependent on conforming preferences but also to the demographic harmony among members, we further introduce utilizing similarities based on demographic characteristics along with the genre-based similarities. We propose applying two distinct strategies for small and large groups by decorating the genre-based similarities with demographic properties, which leads to a more homogeneous automatic group formation. Experiments performed on real-world benchmark datasets demonstrate that each proposed method outperforms its traditional rival significantly, and the final proposed method achieves significantly more qualified ranked recommendation lists than the state-of-the-art algorithm.

1. Introduction

Expert systems are artificial intelligence solutions improving the decision-making process by simulating the knowledge and behavior obtained with human expertise (Duan, Edwards, & Dwivedi, 2019). Recommender Systems (RS), as a variant of expert systems, are intelligent tools in providing qualified referrals that help individuals cope with the enormous amount of information they face through the Internet and support their decision-making process (Turk & Bilge, 2019; Yalcin & Bilge, 2020). They also offer many advantages to online service providers in increasing sales and boosting their popularity (Nunes & Jannach, 2017). Such expert systems accomplish this task by automating the basic human instinct of asking trusted ones for advice and mimic word-of-mouth. Today, they are getting more prevalent with current advances in the communications technology (Felfernig et al., 2019), and present in many areas of daily life as they are talented in discovering engaging content in the Web,[1] social media,[2] digital multimedia platforms,[3] and e-commerce environments.[4]

The bulk of research on RS has been devoted to suggesting relevant products or services to individual users, such as recommending a suitable hotel to a solo traveler based on their desires (Esmaeili, Mardani, Golpayegani, & Madar, 2020). In doing so, it is aimed to satisfy individuals to the maximum extent with the recommended content. However, when the target audience is not an individual, but a group of

* Corresponding author at: Computer Engineering Department, Akdeniz University, 07058 Antalya, Turkey.
E-mail addresses: eyalcin@cumhuriyet.edu.tr (E. Yalcin), abilge@akdeniz.edu.tr (A. Bilge).

[1] https://news.google.com/
[2] https://www.instagram.com/
[3] https://www.netfix.com/
[4] https://www.amazon.com/

https://doi.org/10.1016/j.eswa.2021.114709
Received 3 June 2020; Received in revised form 3 January 2021; Accepted 9 February 2021
Available online 17 February 2021
0957-4174/

users getting together with varying reasons, it is not an easy task to recommend a hotel satisfying all group members or maximizing overall satisfaction (Delic, Neidhardt, Nguyen, & Ricci, 2018; Nguyen & Ricci, April 2018). Thus, Group Recommender Systems (GRS) are introduced as an intelligent system to address the issue of producing referrals appealing to a group of users by aggregating their individual preferences, which requires complicated tools (Masthoff, 2011; Yalcin, Ismailoglu, & Bilge, 2021).

In GRS, user groups are usually identified in four different manners based on their models and forms (Boratto & Carta, 2015). *Random groups* consist of people who are together in a specific moment at the same place by some chance (e.g., people shopping in a mall) (Huang, Xu, Zhu, & Zhou, 2020). On the other hand, people in *occasional groups* unintentionally get together for a specific reason (e.g., people working out in a gym) (Quijano-Sanchez, Recio-Garcia, & Diaz-Agudo, 2011). *Established groups* refer to people who are meant to be together as a community (e.g., members of a family) (Feng & Cao, 2017).

It is common for GRS to produce group recommendations for a set of users that share similar interests rather than providing individual personalized recommendations due to recommendation context or cost (Baltrunas, Makcinskas, & Ricci, 2010). Hence, prior to utilizing a GRS, it is often necessary to build user groups, which are referred to as *automatically identified groups* (Boratto, Carta, & Fenu, 2017; Hurtado, Bobadilla, Gutiérrez, & Alonso, 2020). However, the success of such intelligent systems is strongly correlated to building groups having harmonious users since it is intuitively easier to satisfy like-minded people rather than a randomly ensembled mass. Such automated process of identifying user groups is also helpful since *(i)* manually identifying groups becomes challenging with the increasing number of users and *(ii)* the procedure of grouping users is a continuous process requiring regular updates due to changes in the interests of people over time (Khazaei & Alimohammadi, 2018). Here, the typical approach is to partition users into groups using a clustering algorithm without considering any constraints, and *k*-means clustering technique is the de facto standard due to its efficiency and easy implementation (Boratto et al., 2017; Seo, Kim, Lee, Seol, & Baik, 2018).

1.1. Problem statement

Although automatic identification of groups using conventional clustering approaches is successful in gathering like-minded people together and is prevalent in the literature of GRS (Boratto et al., 2017; Khazaei & Alimohammadi, 2018), it produces groups with an unstabilized number of members. However, some GRS applications might have a constraint about the number of users according to the characteristics of service given. Therefore, identifying groups via traditional clustering methods leads to having a GRS with varying performance on different groups and not always applicable to every group recommendation scenario.

As in traditional RS, the process of clustering users into groups for GRS also suffers from the sparsity problem caused by having too many unrated items in user profiles (Boratto & Carta, 2014; Jeong & Kim, 2019). In the presence of this problem, users are poorly represented by rating vectors, which leads to unreliability in estimating similarities among users. As a result, the clustering algorithm used may fail to identify suitable groups having members with similar tastes, which in turn diminishes overall fulfillment from the GRS employed. Also, proliferating content in GRS complicates the similarity estimation process among users and increases the required computation time, which can be referred to as the scalability problem (Nilashi, Ibrahim, & Bagherifard, 2018; Bilge & Polat, 2013).

Although utilizing personal preferences to build groups sounds like a reasonable method, user ratings per se are inadequate to reflect user profiles. Even if users are determined to be similar in terms of their preferences on items, they can be quite different in terms of demographic attributes such as age range, gender, and occupation (Sridevi & Rao, 2017; Li, Wang, He, Jiao, & Xue, 2017). It follows that users with different demographic attributes may become members of the same group, which violates group homogeneity.

The three concrete application examples below present scenarios in which overall satisfaction from group recommendations is dependent on demographic attributes of users and also there exists a constraint on the size of the constructed groups.

Application scenario 1. Online gaming platforms such as Steam[5] provide gaming rooms for players. For most of the games, there is a restricted number of players and a recommended maximum player count. Therefore, the platform partitions gamers into different rooms to play with others. However, grouping such players according to their gaming tastes, age groups, gender, and occupation, etc. would result in these players enjoying playing together much more, which increases the fulfillment of the gamers since online gaming is a social activity, as well. Also, the platform might recommend unexperienced multiplayer games to players who look for new gaming experiences with unfamiliar but like-minded users. Note that all group members consume the games together as a group in such a scenario.

Application scenario 2. Satellite systems and mobile IPTV service providers are receiving requests for multiple video services with varying bandwidth requirements since some video streams require higher bit rates than others (Maraj, Shehu, Maraj, & Sefa, 2017; Li, Xia, Kang, & Uddin, 2018). Due to bandwidth constraints, the number of viewers in a particular channel must be restricted for ensuring both the quality of service and quality of experience (Boratto, Carta, Chessa, Agelli, & Clemente, 2009). Further, suppose that the service provider presents personalized TV schedule recommendations for channel members as a group, which requires considering personal preferences and demographic characteristics of the members, such as family status and occupational working hours. Note that channel members experience recommendations individually in such a scenario.

Application scenario 3. A travel agency servicing a large number of tourists organizes daily city sightseeing tours to visit several attraction points, with differing characteristics, such as temples, panoramic views, and museums. These tours would be operated by buses with a limited number of seats, which requires subgrouping the customers to form small groups to join the tours together. Furthermore, personalized route recommendations for groups can be produced based on the personal preferences and demographic attributes of the members, such as age groups and education level, to maximize the overall gratification of the group from the tours. Note that all group members experience the tours together as a group in such a scenario.

The limitations emphasized above motivate us to develop novel user grouping approaches, which aim to identify groups that ensure a constrained group size automatically, are resistant to sparsity and curse of dimensionality problems, and sensitive to group homogeneity.

1.2. Contributions and organization

In order to deal with the problems mentioned above, we propose several user grouping approaches building on top of each other. The following summarizes the main contributions of this study.

1. We propose a novel clustering approach based on bisecting *k*-means clustering to automatically identify groups by keeping the maximum size of the groups not exceeding a predefined threshold value. This approach is also helpful in determining the most suitable group for a

[5] https://www.store.steampowered.com/

newcomer, which contributes to the potential scalability problems in large datasets.
2. We adopt an item genres-based profiling approach (Bilge & Polat, 2011) to map large and mostly sparse user-ratings vector into a tiny and fully dense representative vector, which copes with both the sparsity and the curse of dimensionality problems related to similarity estimation process in clustering.
3. We propose two strategies for incorporating demographic attributes of users into the similarity calculation process in order to increase group homogeneity and consequently improve the success of GRS.

The organization of the rest of the study is as follows: The following section presents a brief literature summary on well-known GRS and automatic group identification approaches. Section 3 explains the utilized group recommendation framework. Section 4 introduces the proposed user-grouping approaches in detail, and the following section demonstrates experimental work, obtained results, and gained insights. Finally, Section 6 concludes the study and presents future research directions.

2. Related work

Since the mid-1990s, various GRS have been proposed for different scenarios in several domains such as movies, music, tours, and TV shows. For example, the PolyLens (O'Connor, Cosley, Konstan, & Riedl, 2001) is introduced as an extension of the famous MovieLens and produces movie recommendations for groups of users instead of individual users. The HappyMovie (Quijano-Sanchez et al., 2011) is developed as a Facebook application that recommends movies to groups based on the interests of the group members and the trust among users in the group. Also, the MusicFX (McCarthy & Anagnost, 1998) selects background music for a group of people working out at a gym according to their musical interests. The Adaptive Radio (Chao, Balthrop, & Forrest, 2005) and the FlyTrap (Crossen, Budzik, & Hammond, 2002) are the other popular GRS that are proposed to generate music recommendations. Furthermore, an adaptive GRS called INTRIGUE (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) is proposed to support the organization of guided tours, which aims to produce group recommendations on touristic attraction points by considering the characteristics of tour participants. In addition, the CATS (McCarthy et al., 2006) and the Hootle+ (Alvarez & Ziegler, 2016) are other examples of GRS to support touristic activities.

Also, many advanced GRS have been developed to produce qualified group recommendations in recent years. For example, (Seo et al., 2018) consider the deviations of the preferences provided by group members as an essential element in the aggregation process and combine it with approval voting and average methods to achieve group recommendations. Similarly, (Yalcin et al., 2021) offer to use entropy calculation to analyze the rating distributions during the aggregation process and regard items with high-entropy as not recommendable for the group. Also, some studies consider social relationships and the influences of group members as significant elements in group recommendations (Nozari et al., 2020; Wang, Liu, Lu, Xiong, & Zhang, 2019; Yalcin & Bilge, 2020). For instance, IBGR (Nozari et al., 2020) is an enhanced approach recommending movies to groups by weighting the members' preferences with the similarity and the trust among them. Similarly, TruGRC (Wang et al., 2019) incorporates trust among group members into the aggregation of user preferences. (Huang et al., 2020) employ multiattention-based deep neural network structures to discover internal social features for groups and learn groups' preferences on items to be used in producing group recommendations. Finally, (Wang, Tan, & Goh, 2020) also utilize an attention mechanism and deep neural networks to produce the attention preference weights for members in the group and then use them to provide group recommendations.

The automatic group identification process has been of interest to recent studies in GRS. Baltrunas et al. (Baltrunas et al., 2010) propose identifying groups exceeding a certain intragroup similarity threshold by calculating correlation among group members. Afterward, they produce top-*N* recommendations to groups by ranking items based on predictions produced via a matrix factorization-based collaborative filtering algorithm. However, the common criticism of calculating similarities among all users is the difficulty of finding co-rated items between users and high computation time, especially when applied to a large dataset (Bilge & Polat, 2013). Also, graph clustering approaches (Wang & Fleury, 2011; Fortunato & Castellano, 2012), hierarchical clustering (Lancichinetti, Fortunato, & Kertész, 2009), and modularity-based methods (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008; Newman & Girvan, Feb 2004) are used to identify user communities with similar interests. Fatemi and Tokarchuk (Fatemi & Tokarchuk, 2012) comparatively analyze some of these methods ((Lancichinetti et al., Mar 2009; Wang & Fleury, 2011; Blondel et al., Oct 2008)) and demonstrate that most successful results are achieved by the Louvain, which is a modularity-based community detection algorithm. Specifically, the Louvain algorithm operates a network that is generated based on the similarities between users in order to generate a tree containing hierarchical partitions of the users of increasing granularity in communities, called a dendrogram. Also, there exist some studies in the literature of GRS that utilize the Louvain algorithm to identify user groups automatically (Boratto et al., 2009; Boratto & Carta, 2011; Fatemi & Tokarchuk, 2013).

However, Boratto, Carta, and Satta (2010) comparatively analyze the Louvain and the *k*-means clustering algorithms, and conclude that utilizing the latter performs more successful in detecting appropriate user groups. In this study, they initially predict missing ratings for individuals via a traditional CF algorithm, and then apply *k*-means clustering algorithm on the full user-item matrix containing both user preferences and the predicted ratings in order to identify groups of users with similar interests. This algorithm is referred to as *Predict&Cluster*, and employed in (Boratto & Carta, 2014 & Boratto & Carta, 2015) to demonstrate its superiority comparison to employing *k*-means algorithm on the original user-item matrix for automatically identifying groups.

(Boratto & Carta, 2015; Boratto, Carta, Fenu, Mulas, & Pilloni, 2016) verify the effectiveness of the *Predict&Cluster* method for different group recommendation approaches and aggregation techniques, which are explained in the following section. However, the required time to predict missing ratings in the *Predict&Cluster* method dramatically increases as the content in the domain proliferates. Also, predicting unobserved preferences suffer from sparsity issues since the success of prediction algorithms is strongly depends on the number of genuine ratings in user profiles. To overcome such limitations, (Hammou, Lahcen, & Mouline, 2019) construct an item feature-based matrix that is highly smaller than the complete user-item matrix and utilize it to perform for both predicting the missing ratings and identifying user groups. Although this approach can achieve high-quality group recommendations, it is ignorant of any constraint on group size, as both in (Boratto & Carta, 2015 & Boratto et al., 2016). Thus, these approaches are not always applicable to every group recommendation scenario, as exemplified in the previous section.

Also, there exist some recent studies addressing sparsity issues by completing missing ratings with link prediction on graphs (van den Berg, Kipf, & Welling, 2017) or constructing user-item pairs of nodes with the Vivaldi synthetic network coordinates system (Papadakis, Panagiotakis, & Fragopoulou, 2017). Although these approaches may be beneficial in clustering the users into groups, they construct rating vectors whose size still relies on the number of ratable items. Therefore, the size of such rating profiles increases as content in the domain proliferates, leading to high-computation time problems.

Rather than employing user-ratings vector, latent factors extracted by matrix factorization are also used to represent users in the automatic group identification via *k*-means clustering (Shi, Wu, & Lin, 2015; Liu et al., 2016). Alternatively, Cantador and Castells (Cantador & Castells, 2011) propose constructing user groups by performing hierarchical clustering on ontology-based profiles. Khazaei and Alimohammadi (Khazaei & Alimohammadi, 2018) introduce an automatic user-grouping model based on a modified k-medoids clustering technique by implicit extraction of user preferences/information from their profiles in location-based social networks. Although this approach is effective in partitioning users into groups with a given size, it only becomes suitable when detailed spatial information (e.g., spatial proximity of users, free days of users, and social relationships among them) about users is provided.

Group size plays a vital role in managing group construction and to enhance user fulfillment. However, the vast majority of existing approaches are ignorant of any constraints on group size. There also exist studies addressing sparsity problems in the context of GRS (Khazaei & Alimohammadi, 2018; Boratto et al., 2017); however, there is still a need to improve the scalability of these solutions. Finally, grouping approaches usually consider rating vectors that fall short of representing users decently. Therefore, user representation can be leveraged by extra available information on users.

We conclude this review by noting that *Predict&Cluster* comes into prominence as the state-of-the-art method for the automatic group identification problem, and we utilize it as the benchmark algorithm for evaluating the efficiency of the proposed methods in the present study.

3. Background on utilized group recommendation approach

GRS aims to satisfy not just a sole user but all group members by given recommendations produced by one of the following three approaches (Boratto et al., 2016; Villavicencio, Schiaffino, Diaz-Pace, & Monteserin, 2019). *AggregatedPredictions* approach first predicts the missing ratings of group members by a prediction algorithm and then aggregates both the actual and predicted ratings of members to achieve group ratings on items. Finally, it produces a top-N item list for the group by sorting estimated group ratings in descending order. Similar to the *AggregatedPredictions*, *MergedRecommendations* approach also predicts missing ratings of group members firstly. Then it generates top-N lists for each member individually and aggregates them to having group recommendations. On the other hand, *AggregatedPreferences* approach constructs a group profile by aggregating individual preferences explicitly provided by each group member and then employs it to produce group recommendations.

Several independent studies including (Boratto & Carta, Oct 2015 & Amer-Yahia et al., Amer-Yahia, Roy, Chawlat, Das, & Yu, 2009) have experimentally demonstrated that employing *AggregatedPredictions* approach often produces the most successful group recommendations, which led us to opt for this approach in the present study. Nevertheless, as stated in Section 1, in the vast majority of GRS scenarios, groups are not present in the given preference collection. Thus, before employing *AggregatedPredictions* approach, one first needs to reveal the groups. Therefore, in the following subsections, we give background information on the steps of the group recommendation scenario as depicted in Fig. 1: (i) how to automatically identify groups by k-means algorithm, (ii) producing predictions for individual users, and finally (iii) aggregating predictions to produce group recommendation. Also, Table 1 introduces the abbreviations used in the rest of the study.

3.1. Automatic group identification

To make recommendations of a GRS more appealing to all group members, it is of crucial importance to detect groups appropriately. This task can be considered as a form of clustering problem; thus, it can be accomplished using a fine-tuned clustering algorithm. In the literature, the most popular algorithm used in GRS is the k-means clustering algorithm due to its simplicity and efficiency (Boratto et al., 2017; Seo

Table 1
Abbreviations.

Notation	Description
q	Target item
a	Active user
$\hat{r}_{a,q}$	Predicted rating for user a on item q
$r_{a,i}$	Actual rating of user a on item i
$\overline{r_a}$	Average ratings of user a
$I_{a,u}$	Set of co-rated items of user a and u
U_a	Set of Neighbors of user a
$w_{a,u}$	Similarity value between a and u
AVG	Average
COP	Copeland rule
LM	Least misery
BKM	Bisecting k-means clustering-based approach
BDT	Binary decision tree
GBP	Genre-based profiles
$U_{m\times n}$	Pure ratings-based vectors of users
P	Maximum group size
PBP	Possession-based user-genre profile
RBP	Rating-based user-genre profile
DBP	Demography-based user profile
$w(GBP)_{ab}$	GBP-based similarity between user a and b
$w(DBP)_{ab}$	DBP-based similarity between user a and b
MLP	MovieLens 100 K dataset
MLM	MovieLens 1 M dataset
ML10M	MovieLens 10 M dataset
nDCG	Normalized discounted cumulative gain
P&C	Predict&Cluster
mul	Multiplicative
aug	Augmentative
ICC	Intra-Cluster Correlation

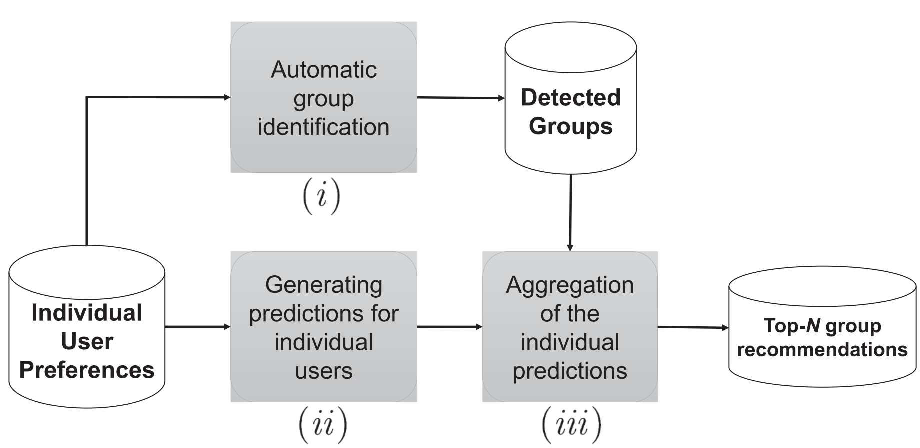

Fig. 1. Group recommendation procedure by *AggregatedPredictions* approach.

et al., 2018; Yalcin et al., 2021). The k-means algorithm takes user-item rating matrix as the input, it then outputs k groups of users with similar interests. Note also that the k-means clustering algorithm employs the adjusted-cosine metric as it is commonly assumed to be the most accurate metric in calculating similarities between two users (Schafer, Frankowski, Herlocker, & Sen, 2007).

3.2. Producing predictions for individual members

As summarized above, one of the main steps in the *AggregatedPredictions* approach is to determine individual predictions for all items that are not evaluated by the group members. These ratings can be produced by a user-based collaborative filtering algorithm (Herlocker, Konstan, Borchers, & Riedl, 1999), which is widely used in the traditional recommender systems. This algorithm estimates $\widehat{r}_{a,q}$ in two steps: (i) locating neighbors by computing similarities between a and the other users and (ii) calculating a prediction as a weighted average of the preferences of neighbors on q. These similarities between users are calculated using various similarity measures. Choi and Suh (Choi et al., Jan. 2013) compared these measures and demonstrated that the most effective measure in the context of collaborative filtering is the Pearson's correlation coefficient, which is given in Eq. (1).

$$w_{a,u} = \frac{\sum_{i\in I_{a,u}}\left(r_{a,i}-\overline{r_a}\right)\left(r_{u,i}-\overline{r_u}\right)}{\sqrt{\sum_{i\in I_{a,u}}\left(r_{a,i}-\overline{r_a}\right)^2}\sqrt{\sum_{i\in I}\left(r_{u,i}-\overline{r_u}\right)^2}} \tag{1}$$

Having computed the similarities, the most similar users are selected from those who rated q as neighbors of a, which forms the set U_a. Then, $\widehat{r}_{a,q}$ for a on the q is estimated using the formula given in Eq. (2).

$$\widehat{r}_{a,q} = \overline{r_a} + \frac{\sum\limits_{u\in U_a}\left(r_{u,q}-\overline{r_u}\right)\times w_{a,u}}{\sum\limits_{u\in U_a}\left|w_{a,u}\right|} \tag{2}$$

3.3. Aggregating individual predictions

After determining groups and estimating individual predictions, a group model that represents the preferences of the group is constructed for each group by aggregating both produced predictions and individual ratings provided by the members. To this end, a large variety of mathematical approaches have been introduced, which are known as the aggregation techniques (Seo et al., 2018). These techniques are divided into three main categories based on their aggregation strategies, which are explained in detail below (Felfernig, Stettinger, Boratto, & Tkalcic, 2018). Note that there is not an optimal aggregation strategy that exhibits the best performance in all application scenarios (Seo et al., 2018). We, therefore, utilize one technique for each strategy in this study. Also, this enables us to analyze how the aggregation strategy influences the performance of the proposed user grouping approaches.

Consensus-based strategy: It addresses aggregation techniques that estimate the group ratings by taking into account the ratings of all members of the group. For this strategy, we select the *Average* (AVG) technique, which estimates the group ratings by simply calculating the averages of the individual ratings (Quijano-Sanchez et al., 2011; Seo et al., 2018). To show how the AVG technique works, we give a user-item matrix in Table 2, which presents recommendations for a group with three members for 6 items on a 5-star scale.

Table 2
Group recommendations by AVG technique.

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	5	1	1	1	5
u_2	2	5	4	2	3	5
u_3	1	2	4	1	2	3
Group (AVG)	1.7	4	3	1.3	3	4.3

Majority-based strategy: The aggregation techniques using this strategy focus on producing a list of items, which are the most popular in a group. For this strategy, we select the Copeland Rule (COP) technique, which extracts the most preferred items by considering the relative importance of the items according to the group members' ratings (Masthoff, 2015). As an example, according to the ratings given in Table 2, the COP aggregates individual ratings per item as in Table 3.

Each element in Table 3 represents the mutual preference status of two items compared to each other. For example, when comparing item i_1 and item i_2 together, i_1 is scored with -1 and i_2 scored with $+1$ since a greater number of members prefer i_2 over i_1. All pairwise comparisons among the items are performed in the same way and the final COP scores are calculated as the sum of the scores obtained from the pairwise comparisons. Finally, recommendations are produced according to the final COP scores.

Borderline strategy: It represents the aggregation techniques that consider only a subset of the group members' ratings. For this strategy, we select the Least Misery (LM) technique in which the lowest rating given by the group members to an item is selected as the group recommendation (Yalcin et al., 2021). An example of the LM technique is presented in Table 4.

4. Novel BKM-based approaches to detect groups automatically

In this section, we present our novel approaches to detect appropriate groups of users. Firstly, we introduce how to construct a binary decision tree (BDT) to be utilized for partitioning users into groups via applying a bisecting k-means clustering algorithm. Then, we explain how to improve the performance of the proposed method by utilizing genre-based profiles (GBP). These profiles are generated by mapping rating-based vectors of users onto genres-based ones to get rid of the adverse effects of the sparse nature of the user-item matrix and to improve the separation skills of the clustering algorithm used. Finally, we present how to integrate these GBP-based similarities with demographic correlations to capture the relations between users in a more accurate way, which in turn leads to constructing more homogeneous groups.

4.1. Constructing BDT via BKM

Clustering algorithms typically take n objects and divide them into k clusters (groups) that have high intra-cluster and low inter-cluster similarities. In order to enhance the performance of GRS, these algorithms have been widely utilized as an off-line process to detect groups of similar users automatically. Also, it enables us to determine the group of which a new user subsequently belongs. One of the most used algorithms in this sense is k-means, which clusters users into the predefined number (k) of groups (Boratto et al., 2017). This algorithm initially

Table 3
Group recommendations by COP technique.

	i_1	i_2	i_3	i_4	i_5	i_6
i_1	■	+1	+1	−1	+1	+1
i_2	−1	■	−1	−1	−1	+1
i_3	−1	+1	■	−1	−1	+1
i_4	+1	+1	+1	■	+1	+1
i_5	−1	+1	+1	−1	■	+1
i_6	−1	−1	−1	−1	−1	■
Group (COP)	−3	+3	+1	−5	−1	+5

Table 4
Group recommendations by LM technique.

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	5	1	1	1	5
u_2	2	5	4	2	3	5
u_3	1	2	4	1	2	3
Group (LM)	1	2	1	1	1	3

selects k random users as cluster centers and then calculates similarities between all users and each cluster center. After that, it assigns each user to the nearest cluster and recalculates the cluster centers. This process repeats until cluster centers stop changing significantly. However, it is a known phenomenon that the performance of the k-means decreases as the number of clusters increases due to high computation time (Bilge & Polat, 2013).

BKM, on the other hand, is a hierarchical clustering algorithm, which is usually utilized in scrutinizing a large amount of data such as recommender systems (Bilge & Polat, 2013), image processing (Zhao, Li, & Cang, 2015) and document clustering (Zhao, Deng, & Ngo, 2018). Initially, this algorithm considers all users as a single cluster and split them into two sub-clusters using the k-means algorithm, which also known as the bisecting step. This process repeats recursively until either desired number of clusters or a particular criterion is reached. Bisecting k-means algorithm has advantages over the standard k-means algorithm: (i) It is more efficient especially when the number of clusters is vast, and (ii) it divides users into clusters with close sizes; while the size of the clusters produced by the standard k-means algorithm varies greatly, which results in clusters being incomparable.

Clustering algorithms are a useful approach to determine groups of similar users. While they are very successful in partitioning users into groups relying on a criterion, their best performance is achieved by partitioning a set of users into two groups because it is more unlikely to occur close user membership values in the case of two groups, which improves the sensitivity of the algorithm (Bilge & Polat, 2013). Therefore, we propose a bisecting k-means clustering-based approach (BKM) to detect groups of similar users, which recursively bisects users into groups and building a BDT according to grouping results. This approach produces leaf nodes in which tiny groups consisting of very similar users and provides to identify which group a newcomer belongs to through simply traversing down the BDT.

Algorithm 1: BKM formation via bisecting k-means clustering (Bilge and Polat, 2013)

```
function BKMP, U_{m×n}    ▷BKM-based grouping
Initialize:
    idx(n)←null    ▷forwarding paths as either 'right' or 'left'
    BDT.centers←null    ▷Group centers
    BDT.left←null    ▷Left sub-tree users
    BDT.right←null    ▷Right sub-tree users
    BDT.RST←anewBDT    ▷Left sub-tree
    BDT.LST←anewBDT    ▷Right sub-tree
Grouping:
    [idx,BDT.centers] = k-means2, U    ▷Bisecting step
    for all user u_i in U (i←1 to m)do
        if idx(u_i) = 'left'
            add u_i into BDT.LST
        else
            add u_i into BDT.RST
        end if
    end for
    if size(BDT.right) > Pthen
        BDT.RST = BKMP,BDT.right
    end if
    if size(BDT.left) > Pthen
        BDT.LST = BKMP,BDT.left
    end if
    return BDT
end function
```

Given pure rating-based vectors of users ($U_{m\times n}$) in a user-item rating matrix and maximum group size (P), the BKM algorithm works as follows. At each level, it divides users into two distinct groups by running k-means on the U, which is so-called the bisecting step. In the meantime, the centers of these groups are also recorded to be used in determining the group of a newcomer. In case the number of users in a group exceeds P, then the BKM algorithm is called recursively in an attempt to bisect that particular group. This process repeatedly continues until all groups have P users at most; thus, the value of P can be considered as a stopping criterion. Finally, a BDT having recorded group centers as branch nodes and groups of similar users at leaf nodes is constructed. Algorithm 1 summarizes the BKM. Note that, similar to k-means clustering, BKM also utilizes the adjusted-cosine similarity metric to compute similarities between user vectors (Schafer et al., 2007).

While in the case of standard k-means, all users and all cluster centers are involved in similarity calculation at each iteration, only users of one group, as well as two group centers, are considered at each bisecting step in the BKM, which reduces the computation time significantly. In addition, once a BDT has been constructed, the group to which a new user belongs is determined by traversing the BDT top-down. At each level, two similarity values are calculated, one for each group center. The node that corresponds to the higher similarity value is selected as the next node, which is stored in a variable defined as *idx* in Algorithm 1. Hence, the final group of the new user is determined with at most $2\times(H-1)$ similarity calculations, where H denotes the height of the BDT. Although H usually depends on the total number of users m, it is naturally much less than m. Also, intuitively, H is much smaller than the total number of identified groups k especially when the system suffers from the scalability problem. Consequently, the group of a newcomer can be identified after at most $2\times(H-1)$ similarity calculations instead of either m (Baltrunas et al., 2010) or k, which reduces the overall computation time.

An example of a BDT formed by the BKM algorithm is presented in Fig. 2. Initially, the number of users in the dataset is 233, and P is selected as 20 in this example. At the first level, the users are partitioned into two groups with 118 and 115 users. Group centers are recorded at the bottom side of the root node, as C_1^R and C_1^L, where subscripts denote the height of the current BDT which is incremented by one at each next level. Also, the superscripts show for which sub-tree the group center forwards (right or left). Likewise, in the right sub-tree of the root node, 115 users divided into two groups with 53 and 62 users with the branch node, including C_2^R and C_2^L. All branch nodes are partitioned repeatedly into two groups until they include at most P users. In doing so, we end up with leaf nodes that correspond to the final groups. For example, the rightmost leaf node with index 16 includes 14 users and is denoted as G_{16}. Finally, BDT is constructed with two group centers at each branch node to enable forwarding procedure and small user groups with the same sizes at each leaf node to facilitate identifying groups appropriately. In this example, detecting the group which a new user belongs to requires at most $2\times(4-1)=6$ similarity calculations instead of 16 (which is k in this case), decreasing the number of calculations approximately three times. Therefore, it is clear from the example that grouping users with the BKM algorithm overcome the scalability problem while identifying groups with a threshold size.

4.2. Improving the BKM via GBP

Computation of similarities between user profiles in a robust way is the essential part of identifying groups of similar users. This is normally built upon items that are commonly rated, as stated in Eq. 1. However, as the number of items increases, user profiles become much more sparse, which makes difficult to obtain co-rated items. Even if there exist some co-rated items, often the number of them is not sufficiently great to calculate the similarities robustly. Also, computation time required to estimate similarities increases as the number of available items grows which causes the problem of high-dimensionality.

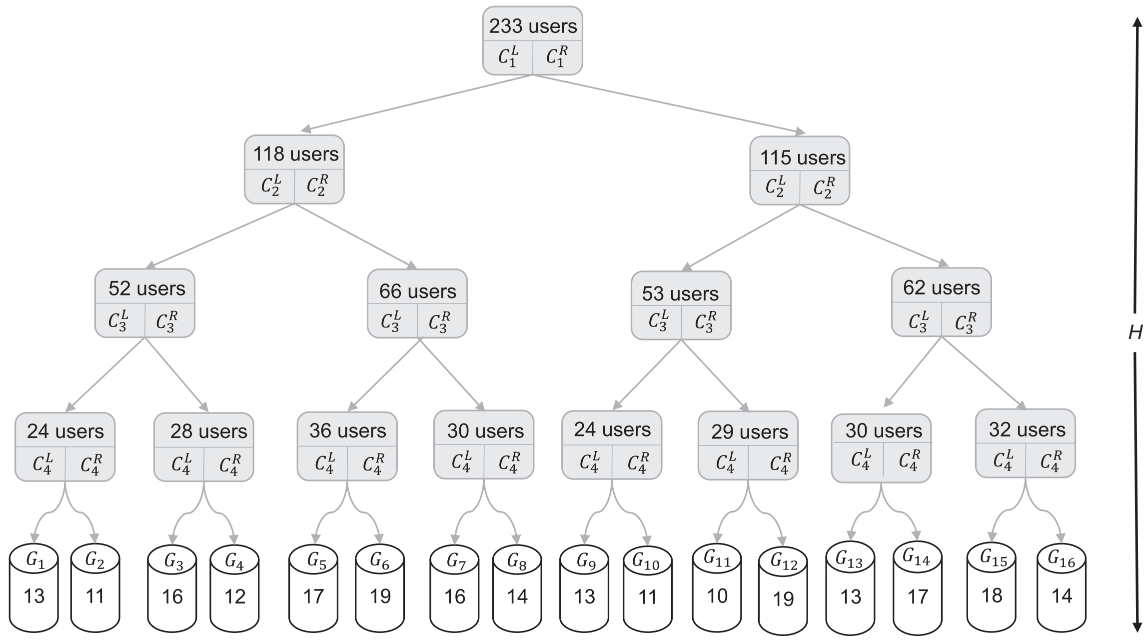

Fig. 2. An example of a BDT formed via BKM.

To deal with the sparsity and scalability issues, user profiles can be composed of the genres of the items that are preferred by the users, as in (Bilge & Polat, 2011). Here, the main goal is to map the original rating-based vectors of users U that are sparse and large, into genre-based profiles (GBP) that are dense and much smaller. This enables one to calculate similarities among users even in the absence of co-rated items. To create GBP, it is necessary to categorize items according to their genres. For example, these genres could be horror, drama or comedy for movies; pop, rock or classical for music, and autobiographies, romance or mystery for books. Additionally, since the number of genres is a constant value and much more smaller than the number of items, the required time to calculate similarities becomes stabilized and reduced. As a result, instead of using U, we adopt two user profiling approaches that generate GBP to boost the performance of the proposed BKM-based approach. We explain these approaches below.

Possession-based user-genre profile (PBP): PBPs are produced by checking whether an item is rated or not. If an item is rated, then the corresponding genres of the corresponding item are incremented. Note that this profiling approach only considers whether an item is possessed or not, thus disregards the magnitude of the liking degree.

Rating-based user-genre profile (RBP): RBPs are produced by firstly checking whether an item is rated or not as the initial task. In case an item is rated by the user, the corresponding genre categories of the item are increased by the magnitude of the rating. In other words, if a user provides an opinion on an item by rating it, then each genre category of that item is increased by as much as the value of the rating. Note that this profiling approach not only checks whether an item is possessed or not but also considers how much the user likes the item.

To understand how PBP and RBP work, a small user-item matrix is given in Table 5, containing ratings of two users for five books. Here, the ratings are discrete and on a 5-star scale; and $\perp$ denotes the unrated items. Also, each book belongs to at least one genre from the set: {*Classic, Legend, Fantasy, Mystery, Romance*}, as shown in Table 6. If a book belongs to a genre, then the corresponding cell is 1, and 0 otherwise.

According to the example given in Table 5 and Table 6, corresponding PBPs and RBPs are depicted in Fig. 3. As seen in Fig. 3, the PBP of Alice is generated as $[1, 2, 1, 1, 1]$ for genres *Classic*, *Legend*, *Fantasy*, *Mystery*, and *Romance*, respectively. Concretely, there are three items, b_1, b_2, and b_5, whose genre is *Classic* and she provided a rating for only one of them; thus, corresponding PBP value is 1. Likewise, there are three items, b_2, b_4, and b_5, whose genre is *Legend* and she rated two of them; thus, corresponding PBP value is 2, and so on. Similarly, the PBP of Peter is generated as $[2, 1, 1, 1, 1]$. When it comes to RBP, the RBP of Alice and that of Peter are generated as $[2, 6, 5, 5, 4]$ and $[7, 3, 3, 4, 4]$, respectively. Among the items rated by Alice, only b_2 is a *Classic* book; thus, the rating given by Alice for that book forms the value of *Classic* in her RBP. Also, in the category of *Legend*, Alice rated b_2 and b_4. The sum of the ratings given by Alice for these movies is equal to $2 + 4 = 6$, which forms the value of *Legend* in her RBP. The same applies to the remaining genres.

Having generated GBPs, we also normalize them for two reasons. First, the number of rated items may differ from user to user, and second, each item may belong to different numbers of genres. For example, as in Table 5, Alice provides ratings for three items, while Peter rates two items. Also, b_1 belongs to three genres while b_3 belongs to two genres, as seen from Table 6.

To tackle these problems, we normalize the GBPs dividing each value

Table 5
An example of user-item matrix.

	b_1	b_2	b_3	b_4	b_5
Alice	$\perp$	2	5	4	$\perp$
Peter	4	$\perp$	$\perp$	$\perp$	3

Table 6
Genres of books.

	Classic	*Legend*	*Fantasy*	*Mystery*	*Romance*
b_1	1	0	0	1	1
b_2	1	1	0	0	0
b_3	0	0	1	1	0
b_4	0	1	0	0	1
b_5	1	1	1	0	0

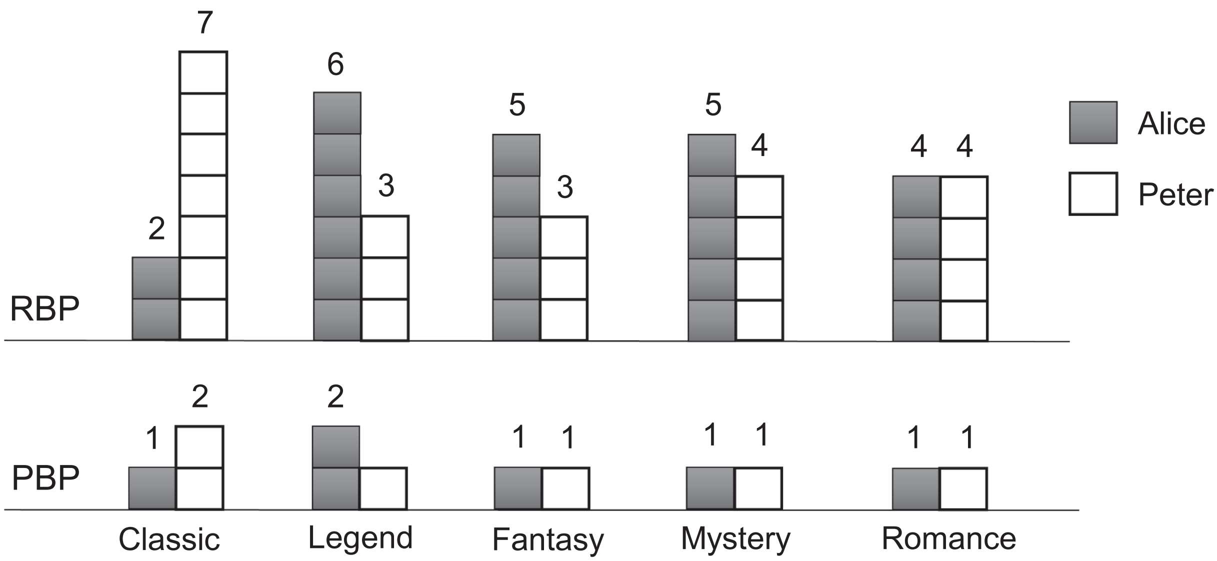

Fig. 3. Genre-based profiles for Alice and Peter.

in a profile by their sum. Hence, Alice's GBPs are normalized as $\left[\frac{1}{6},\frac{2}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6}\right]$ and $\left[\frac{2}{22},\frac{6}{22},\frac{5}{22},\frac{5}{22},\frac{4}{22}\right]$ for PBP and RBP, respectively. Similarly, Peter's GBPs are normalized as $[\frac{2}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6}]$ and $\left[\frac{7}{21},\frac{3}{21},\frac{3}{21},\frac{4}{21},\frac{4}{21}\right]$ for PBP and RBP, respectively.

After constructing GBPs of all users, similarities are calculated on these tiny profiles. Also note that in this example, similarity between Alice and Peter can be calculated via their GBP, even though there are not any items that are co-rated by them.

4.3. Decorating GBP-based similarities with demographic correlations

In the previous section, we proposed to use GBP to construct user profiles for GRS, which comes with several advantages over the traditional rating vectors. However, GBP still rely only on the user's preferences on items. Hence, they do not reflect any other information about users, leading to poor representation of users and consequently having heterogeneous groups whose members will not be equally satisfied with offered recommendations. To address this issue, we propose to incorporate demographic information-based correlations of users along with their GBP-based similarity. In doing so, the demographic correlation between users plays a role in calculating user-user similarity, leading to having more homogeneous groups.

We propose two demographic–correlations based approaches. In both approaches, the first step is to extract demographic categorical vectors of users from their available demographic information. In the case of MovieLens dataset, such information consists of age, gender, and a choice from a set of 21 possible occupations for each user in the data set.

Before calculating similarities based on demographic attributes, first, it is needed to generate demography-based user profiles (DBP) having 27 features, as explained in Table 7. Having generated DBP along with GBP (RBP or PBP) for all users, we calculate two types of similarities for each pair of users; one is based on GBP and the other is based on DBP. We denote GBP-based similarity between user a and user b with $w(GBP)_{ab}$; and for DBP-based similarity with $w(DBP)_{ab}$. To estimate the ultimate similarities combining $w(GBP)_{ab}$ and $w(DBP)_{ab}$, we propose two different approaches explained below.

Multiplicative: This approach multiplies $w(GBP)_{ab}$ by $w(DBP)_{ab}$ to obtain the ultimate similarity value, denoted as w_{ab} and formulated in Eq. (3). Note that the *multiplicative* approach utilizes demographic correlations as a factoring coefficient on preference-based similarity values.

$$w_{ab} = w(GBP)_{ab} \times w(DBP)_{ab} \tag{3}$$

Augmentative: This approach utilizes genre-based similarities as the driving force, while demography-based correlations as an additive factor on the ultimate similarity value, and formulated in Eq. (4). Note that, the *augmentative* approach values preference-based similarities as the decisive factor, while it employs demographic correlations as an auxiliary influencer.

$$w_{ab} = w(GBP)_{ab} + \left[w(GBP)_{ab} \times w(DBP)_{ab}\right] \tag{4}$$

Table 7
The structure of demographic profile.

#Feature	*Name*	*Contents*	*Explanations*
1	Age	*Age* ⩽*18*	each user belongs to only one age category, the corresponding cell is assigned as 1, and the rest of them assigned as 0.
2		*18* ⩽*Age* ⩽*29*	
3		*29* ⩽*Age* ⩽*49*	
4		*49* ⩽*Age*	
5	Gender	*Male*	cell referring to the user gender is assigned as 1, and the other cell is assigned as 0.
6		*Female*	
7	Occupation	*administrator*	a single cell referring to user occupation is assigned as 1, and the rest of the cells are assigned as 0.
8		*artist*	
9		*doctor*	
⋮		⋮	
27		*writer*	

Employing PBP/RBP reduces the size of user vectors dramatically, which accelerates the automatic group identification process with the proposed BKM approach. Construction of PBP/RBP requires $\mathscr{O}(n)$ complexity (n: #items), and inclusion of demographic info requires $\mathscr{O}(1)$ complexity since it introduces a constant number of algebraic operations. However, utilizing PBP/RBP in the clustering process rather than original user vectors reduces the clustering cost dramatically since the number of genres is much smaller than ratable items. Thus, our proposed methods help mitigate the adverse effects of the scalability issue related to the similarity estimation process in clustering.

5. Experimental studies

In order to scrutinize the effectiveness of the proposed grouping approaches, we have conducted several experiments on real-world datasets.

5.1. Datasets and evaluation metrics

The famous MovieLens dataset, which is collected by the GroupLens research team at the University of Minnesota[6], is employed in the experiments. MovieLens has three versions according to the number of ratings included, i.e., 100 K (MLP), 1 M (MLM) and 10 M (ML10M), whose detailed information is presented in Table 8. In these datasets, there are 18 movie genres to categorize the movies, and each movie belongs to at least one of them. Besides, the demographic information about users, such as their age, gender, and occupation, is also available in MLP and MLM, which allows for scrutinizing the effects of the proposed approaches.

In order to measure the performance of the proposed approaches on top-N group recommendation, we utilized the normalized Discounted Cumulative Gain (nDCG) metric, which is commonly used in group recommendation research (Seo et al., 2018). Concretely, the nDCG estimates the goodness of a ranked recommendation list by considering both the actual ratings of the items and the positions of the items in the recommendation list.

Suppose that u is a member of a group G, and $r_{u,i}$ denotes the actual rating of u for item i. If $K = \{i_1, i_2, \ldots, i_k\}$ indicates ranked items produced as recommendation list for G, then discounted cumulative gain (DCG) and nDCG for each member in the G are calculated as given in Eqs. (5) and (6), respectively.

$$\mathrm{DCG}_k^u = r_{u,i_1} + \sum_{n=2}^{k} \frac{r_{u,i_n}}{log_2(n)} \tag{5}$$

$$n\mathrm{DCG}_k^u = \frac{\mathrm{DCG}_k^u}{\mathrm{IDCG}_k^u} \tag{6}$$

where IDCG_k^u denotes the maximum possible gain for u that is obtained with the optimal re-ordering of k items. In calculating the nDCG value for each user, we first assign a rating of '0' in place of null values, to the items that were not rated by the user, as in (Gorla, Lathia, Robertson, & Wang, 2013), to penalize the model which recommends items that were not rated by the user. As a result, a GRS that recommends items referred by the users achieves higher nDCG scores.

To evaluate the accuracy of the recommended top-K items, we also employ *Precision*, *Recall*, and *F1-score* metrics (Ha & Lee, 2017), formulated in Eqs. (7)–(9), respectively. Specifically, *Precision* can be defined as the ratio of recommended relevant items to all recommended items. On the other hand, *Recall* measures the ratio of recommended items that are relevant to all relevant items in the user profile. Finally, *F1-score* can be calculated as the harmonic mean of *Precision* and *Recall*. Note that we set the threshold value as 3.5 to determine whether an item is relevant or not while calculating these metrics since the positive ratings correspond to 4 and 5 in the {1,5} rating scale (Bobadilla, Serradilla, & Bernal, 2010).

$$Precision = \frac{\sum_{i \in K} 1\left(i \in p_u\right)}{|K|} \tag{7}$$

$$Recall = \frac{\sum_{i \in K} 1\left(i \in p_u\right)}{|p_u|} \tag{8}$$

[6] http://www.grouplens.org/

Table 8
Descriptions of datasets.

Dataset	#Users	#Items	#Ratings	Sparsity (%)	Rating scale
MLP	943	1,682	100,000	93.70	{1, 5} 5-star
MLM	6,040	3,952	1,000,209	95.75	{1, 5} 5-star
ML10M	71,567	10,681	10,000,054	98.69	{1, 5} 10-star

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

where p_u indicates the set of relevant items in the profile of the user u and $1(.)$ is the indicator function that returns 1 if its argument is true and 0 otherwise.

5.2. Benchmark algorithm: Predict&Cluster

We set the *Predict&Cluster* (*P&C*) method as the benchmark algorithm to evaluate the efficiency of our proposed methods as it is the most prominent approach in the automatic identification of user groups in GRS (Boratto et al., 2010; Boratto & Carta, 2014; Boratto & Carta, 2015; Boratto, Carta, & Fenu, 2016; Boratto et al., 2017), as emphasized in Section 2. *P&C* is an algorithm that is conceived to enhance the clustering process performed for identifying user groups.

Principally, *P&C* identifies groups of similar users by applying the standard k-means clustering algorithm not on the original user-item matrix containing only the ratings of users, but on the full matrix where unrated cells are predicted using the user-based CF algorithm, as stated in Section 3.2. By taking advantage of eliminating the adverse effects of the sparsity problem in clustering, this algorithm can identify better-suited groups, i.e., clusters that consist of like-minded users having similar tastes.

In conclusion, having detected user groups, *P&C* generates group recommendations by aggregating formerly produced individual predictions, as explained in Section 3.3.

5.3. Experimentation methodology

In this study, we followed a 10-fold cross-validation experimentation methodology to evaluate the grouping approaches. To perform the cross-validation, the set of items is uniformly randomly partitioned into ten subsets (each subset contains 10% of the items). At each iteration, one of the subsets is employed as the test set and the combination of the remaining nine subsets is utilized as the training set. Note that overall nDCG values are obtained by taking the average of 10-fold experimental accuracy results.

After the test and training sets are constructed, groups are identified with standard k-means, *P&C*, and BKM variant algorithms based on the training set. To compare the performance of the BKM variants with both k-means and *P&C*, it is first necessary to determine the number of the groups (k) to be constructed for k-means and *P&C*. So, we calculate the value of k with the ratio of the total number of users to the threshold value (P), which indicates the maximum size of the groups in the BKM approach. For example, if the value of P in the BKM is set to 50, and the number of total users in the data set is 600, then the value of k is set to $600/50 = 12$. To assess the impact of the group size on the performance of the proposed approaches, we perform several experiments with varying P values ranging from 5 to 200. Note also that we classified the groups in the experiments as *small* (#members $\leqslant$10), *medium* (10 $<$ #members $\leqslant$100), and *large* (#members $>$ 100) according to their sizes.

After identifying groups, we predicted the individual ratings for each item employing a user-based collaborative filtering algorithm, as described in Section 3.2. Then, we predict ratings of the unrated items for each group based on the three aggregation techniques, namely AVG, LM, and COP. After that, we produce top-N lists for each group based on the group recommendations, where N is set to 5, 10, and 20. Finally, we evaluate the accuracy of the produced top-*N* lists via the *n*DCG metric.

5.4. Experimental results

5.4.1. Effects of BKM algorithm

To examine the accuracy performance of the proposed BKM algorithm on detecting groups of similar users, we performed many trials with different parameters, including the group size, the utilized aggregation technique, and the size of the recommendation list (*N*). Also, we compared empirical outcomes against standard *k*-means for both MLP and MLM data sets, as displayed in Table 9 and Table 10, respectively.

As can be seen in Table 9, the *n*DCG results of the experiments conducted for MLP demonstrate that the BKM algorithm is relatively better than the *k*-means for each different setting. We also performed statistical significance tests to compare obtained results for the BKM and *k*-means, as given in the footnote of tables. The results of the one-tailed *t*-tests claim that improvements of the BKM over *k*-means appear to be statistically significant, especially for small and medium groups. However, even if the BKM performs better than *k*-means for large groups, these differences are hardly significant.

Similarly, the BKM outperforms *k*-means for MLM as it can be seen from Table 10. Although MLM is sparser and immensely larger than MLP, all improvements appear to be statistically significant at 99% confidence level except the case in which the utilized aggregation technique is LM, and top-*N* and *P* are set to 20 and 200, respectively, which is also significant at 95% confidence level. Thus, it can be concluded that the effects of the proposed BKM allow providing high-quality group referrals by detecting more appropriate groups, especially in massive datasets; this also presents the robustness of the BKM in terms of scalability.

As can be followed by Table 9, even if AVG and COP have similar performances for MLP, the highest accuracy results are achieved by employing COP technique. In other words, COP technique performs even better on the aggregation of individual predictions than other ones. Also, experiment results show that the worst accuracy performance is obtained when LM is utilized as the aggregation technique. Besides, while the accuracy performances of AVG and COP improve with increasing group size, it decreases for LM considerably. Such phenomena occur since LM takes into consideration only the lowest rating in the group and disregards the rest of the members' opinions, which in turn leads to provide improper group recommendations, especially for large groups (Seo et al., 2018).

Although COP outperforms other aggregation techniques in the experiments conducted for MLP, its accuracy results drastically decrease in MLM due to sparsity issues, as seen in Table 10. Because the success of its procedure of extracting the most preferred items in the group mainly depends on the dependability of the predicted ratings for unrated items. However, the collaborative filtering algorithm employed for producing predictions also commonly suffer from the sparsity problem, and it leads to producing not reliable predictions. Note also that the required time to compute pairwise comparisons among items while performing COP technique in MLM increases considerably due to the number of available items. Therefore, according to the *n*DCG results of experiments performed for both datasets, it can be concluded that utilizing AVG technique to aggregate individual predictions is generally more effective than the other techniques. The reason for this consequence is that AVG provides a consensus in the group since the effect of each member on group recommendation is equal. Hence, the interests of all members for an item are reflected equally in the group recommendation score. Finally, it can be unsurprisingly concluded that the size of the recommendation list (*N*) has a positive effect on accuracy since the chance of getting a hit increase obviously with a bigger recommendation list.

Table 9
*n*DCG results of *k*-means and BKM for MLP dataset.

	Group Size (*P*)		Small		Medium		Large	
Aggregation Technique	top-*N*	Model	5	10	25	50	150	200
AVG	5	*k*-means	0.159	0.161	0.166	0.167	0.176	0.194
		BKM	0.171†	0.181†	0.190†	0.194†	0.197	0.218*
	10	*k*-means	0.173	0.184	0.194	0.195	0.196	0.188
		BKM	0.185†	0.194*	0.202*	0.213*	0.216	0.222†
	20	*k*-means	0.208	0.218	0.223	0.229	0.230	0.234
		BKM	0.225†	0.223†	0.239†	0.247*	0.250*	0.258†
LM	5	*k*-means	0.146	0.141	0.126	0.115	0.112	0.124
		BKM	0.164†	0.158*	0.162†	0.141†	0.122	0.124
	10	*k*-means	0.158	0.143	0.141	0.125	0.121	0.112
		BKM	0.176†	0.167†	0.156†	0.155†	0.148†	0.127
	20	*k*-means	0.189	0.169	0.150	0.144	0.124	0.127
		BKM	0.199†	0.189†	0.179†	0.170*	0.146*	0.130
COP	5	k-means	0.163	0.162	0.170	0.178	0.198	0.192
		BKM	0.173*	0.177*	0.190†	0.202†	0.224*	0.212*
	10	*k*-means	0.176	0.188	0.188	0.198	0.211	0.236
		BKM	0.189†	0.201†	0.207*	0.227†	0.236*	0.232
	20	*k*-means	0.217	0.220	0.236	0.234	0.240	0.231
		BKM	0.221	0.227	0.244*	0.243	0.257	0.255*

* For significance at 95%.
† For significance at 99%.

Table 10
*n*DCG results of *k*-means and BKM for MLM dataset

	Group Size (*P*)		Small		Medium		Large	
Aggregation Technique	top-*N*	Model	5	10	25	50	150	200
AVG	5	*k*-means	0.139	0.150	0.166	0.172	0.184	0.188
		BKM	0.152†	0.172†	0.197†	0.214†	0.225†	0.221†
	10	*k*-means	0.147	0.152	0.172	0.170	0.187	0.195
		BKM	0.155†	0.171†	0.199†	0.201†	0.223†	0.222†
	20	*k*-means	0.164	0.175	0.190	0.197	0.210	0.222
		BKM	0.171†	0.186†	0.210†	0.220†	0.236†	0.241†
LM	5	*k*-means	0.119	0.113	0.107	0.090	0.082	0.079
		BKM	0.142†	0.144†	0.142†	0.130†	0.117†	0.111†
	10	*k*-means	0.124	0.122	0.110	0.100	0.090	0.086
		BKM	0.142†	0.145†	0.141†	0.129†	0.113†	0.107†
	20	*k*-means	0.136	0.127	0.122	0.111	0.097	0.099
		BKM	0.152†	0.152†	0.147†	0.137†	0.118†	0.110*
COP	5	*k*-means	0.076	0.072	0.059	0.077	0.067	0.051
		BKM	0.085†	0.087†	0.075†	0.101†	0.085†	0.068†
	10	*k*-means	0.067	0.066	0.059	0.071	0.063	0.065
		BKM	0.076†	0.079†	0.073†	0.085†	0.080†	0.079†
	20	*k*-means	0.093	0.087	0.076	0.091	0.087	0.075
		BKM	0.101†	0.099†	0.089†	0.108†	0.101†	0.089†

* For significance at 95%.
† For significance at 99%.

5.4.2. Effects of GBP

In this section, to investigate the effects of employing GBP instead of pure rating-based vectors of users (U) while performing the BKM, we conducted many evaluations with varying parameters. We compare the accuracy performance of utilizing either U or GBPs (both RBP and PBP) using *n*DCG results derived from experiments on both datasets. Also, we visualize the experimental outcomes for MLP and MLM datasets via interaction plots, as depicted in Figs. 4 and 5, respectively.

As seen in Fig. 4, utilizing GBPs outperforms original rating matrix for all settings, and RBPs are more successful in detecting appropriate groups of users in comparison to PBPs, especially for large groups since RBP scheme generate user-profiles by considering not only whether an item is rated or not but also how much users like or dislike certain types of item genres. In addition, with the increasing size of groups, both PBP and RBP significantly enhance the accuracy of detecting groups of similar users, except the case in which group size is 150. Similarly, it can be followed that the performance of GBP and BKM becomes slightly better with the increasing size of the recommendation list.

Similar to the results in the previous section, COP and AVG achieve relatively better results than LM technique, and their performances improve with increasing group sizes, as well. On the other hand, the accuracy of LM technique becomes worse as the group size increases and its performance for large groups significantly worse than COP and AVG.

As shown in Fig. 5, the results of experiments conducted for the MLM indicate that GBPs are helpful generally for obtaining proper user groups. However, since the number of available items in MLM is considerably larger than MLP, RBP and PBP have close accuracy performance, which indicates that with the growing number of profiles and products, both profiling mechanisms converge to a particular improvement level. In addition, with increasing group size in the small and medium groups, all three proposed models enhance the accuracy of identifying appropriate groups of users; however, in large groups, their performances become worse as the size of groups increases.

Also, it can be concluded that the performance of all three proposed models enhances as the size of the recommendation list increases, although there is a slight decrease in top-10 recommendations. Besides, AVG technique performs significantly better than other ones for all three models, as can be seen in Fig. 5. On the other hand, COP performs relatively worse than the other aggregation techniques. Finally, while AVG enhances the quality of the group modeling procedure for larger groups, the performances of COP and LM decrease as the group size increases as previously discussed.

5.4.3. Effects of combining DBP with GBP

We conducted the last set of experiments in order to examine the effects of using demographic correlations on identifying appropriate groups of users. We utilized multiplicative (subscripted as *mul*) and augmentative (subscripted as *aug*) approaches that are decorated on RBP-based similarities in these experiments since RBP performs relatively better than PBP, as demonstrated in the previous section. Also, we present the *n*DCG results of our proposed ultimate algorithms (i.e., *mul* and *aug*) against both RBP and the benchmark algorithm, *P&C*, for MLP and MLM data sets, as seen from Table 11 and Table 12, respectively. Note also that we again performed paired *t*-tests to compare the *n*DCG results and presented the statistical significance levels in the footnote of tables.

According to the results presented in Table 11, for small groups, even if *mul* and *aug* slightly improve the performance of RBP, the obtained enhancements appear not to be statistically significant in all schemes. On the other hand, for medium and large groups, the outcomes demonstrate that *mul* and *aug* is highly effective and significantly improve accuracy. Thus, it can be concluded that as group size increases, utilizing user preferences alone becomes inadequate for estimating similarities between users and identifying more homogeneous groups.

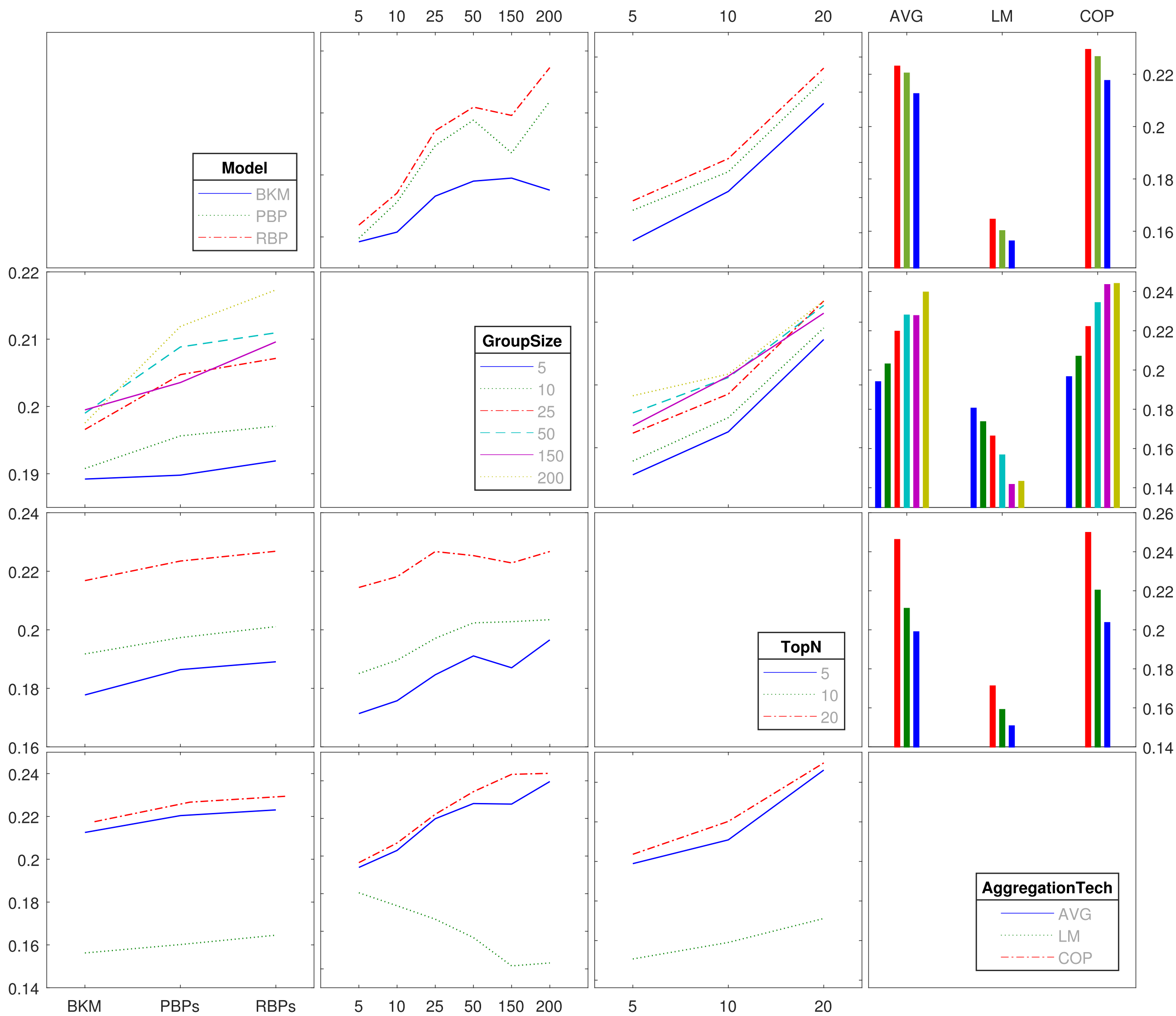

Fig. 4. Effects of GBP for MLP data set.

Although *mul* and *aug* improve the performance of RBP for both MLP and MLM, they perform considerably better for MLM, as can be seen from Table 12. Almost all enhancements appear to be statistically significant, concluding that utilizing demographic correlations contribute to the better grouping of users regardless of the aggregation technique employed and group size.

As can be followed by Table 11, when LM aggregation technique is used, *mul* and *aug* achieve comparable results with *P&C*. However, when Avg or COP is utilized as the aggregation technique, both *mul* and *aug* significantly outperform *P&C*, especially for medium and large groups. In comparing the results obtained from the MLM data set in Table 12, *mul* and *aug* significantly outperform *P&C* for all schemes except for large groups when LM is utilized. Therefore, we conclude that the proposed ultimate automatic group identification methods, i.e., *mul* and *aug*, perform significantly better than the benchmark algorithm when combined with the proper aggregation technique.

5.4.4. Evaluations on computational cost and group formation efficiency

In this section, we perform further experiments for comparing the computational cost and group formation efficieny of our approaches against both *k*-means and benchmark user grouping algorithm *P&C*. In these experiments, we consider all group formations, i.e., small, medium, and large, by selecting *P* as 10, 50, and 200, respectively.

In order to compare computational costs, we measure the overall running times by taking the average running time of the experiments performed in each fold and present them in Table 13. Note that these experiments are conducted in a system with Intel(R) Core(TM) i7-7700HQ 2.80 GHz processor, with 16-GB DDR3 1600 RAM, using Matlab R2017b.

As can be seen in Table 13, the worst running performance is obtained when the *P&C* is utilized since it predicts all missing ratings in the user-item matrix before constructing user groups by *k*-means, which leads to an extra substantial cost. On the other hand, the required time to identify groups by *k*-means is usually lower than other methods, especially for the MLP dataset; however, its accuracy performance is significantly worse than all others, as demonstrated in the previous experiments.

The empicial outcomes show that our BKM approach is relatively better than *P&C* and it becomes much faster when RBP is utilized to estimate similarities, leading to the best outcomes for the MLM dataset. The main reason for this finding is that the number of genres is much smaller than ratable items and size of RBP vectors is dramatically smaller than the original rating vectors. Thus, it significantly accelerates the similarity calculation process. The results also demonstrate that decorating RBP with either *mul* and *aug*, we end up with RBP_{mul} and RBP_{aug} methods, which leads to a slowdown in constructing user groups

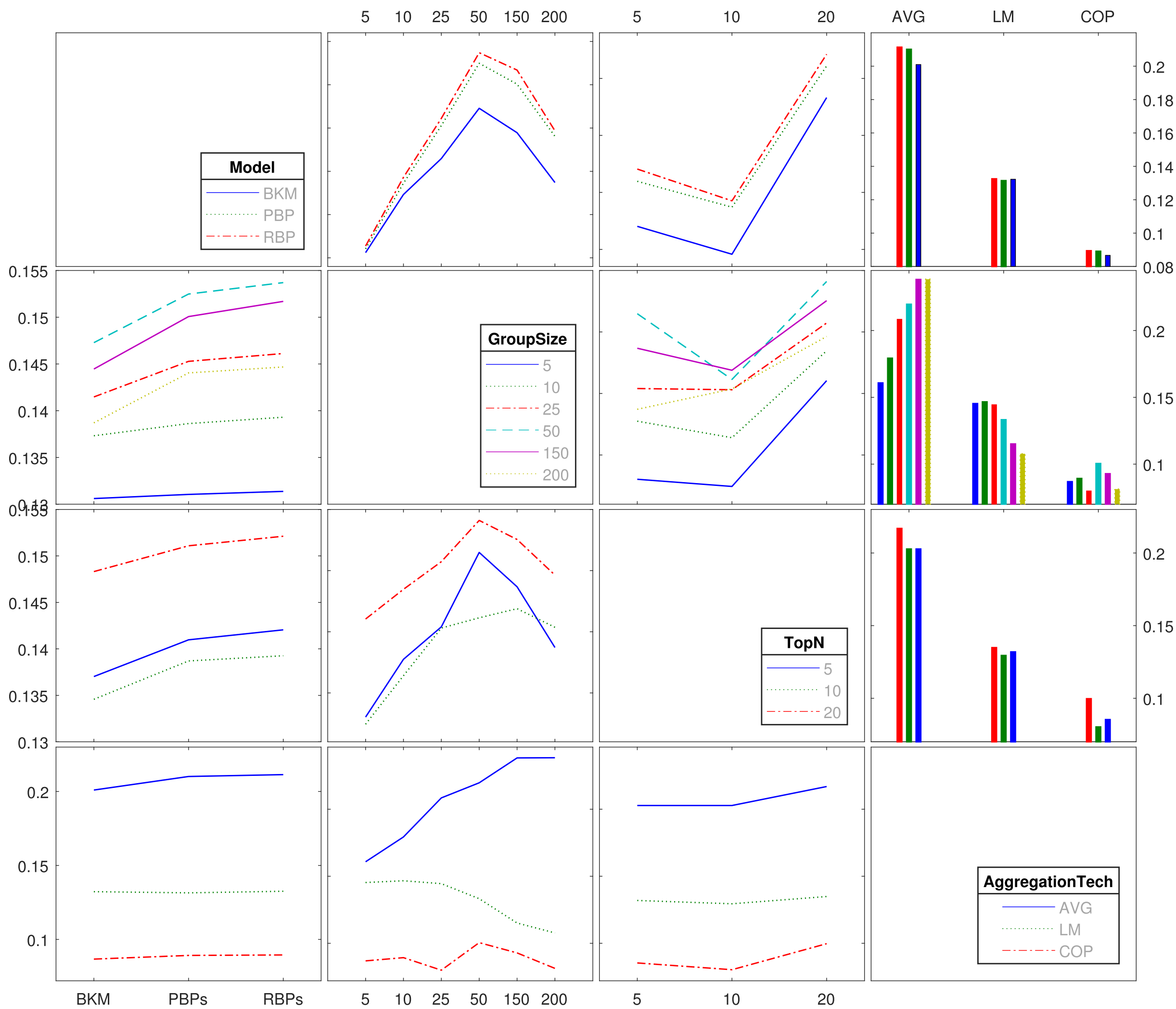

Fig. 5. Effects of GBP for MLM data set.

as demographic correlations are also included in similarity estimation process; however, they are still better than both BKM and benchmark *P&C* methods. In conclusion, it can be concluded that our proposed user grouping methods, especially RBP, significantly scalable than benchmark methods in identifying user groups.

We also compare the proposed approaches' group formation efficiency by presenting Intra-Cluster Correlation (ICC) within groups and the standard deviation of constructed groups' sizes. For a particular group, ICC demonstrates group homogeneity by calculating Pearson's correlation coefficient between all group members and the corresponding group center and taking the average of these values. An overall ICC score is obtained by averaging all ICC scores for constructed groups and presenting how closely related the group members are in overall formations. Therefore, ICC can be considered a measure of group homogeneity, and the higher the ICC, the better the constructed groups. We present ICC scores of *k*-means, *P&C*, our ultimate RBP_{mul} and RBP_{aug} schemes for MLP and MLM datasets in Fig. 6. Note that both *k*-means and *P&C* schemes sometimes construct groups with only one member, and we do not consider groups with less than two members since such outliers lead to misleading results.

As can be followed from Fig. 6, both of the proposed approaches, i.e., RBP_{mul} and RBP_{aug}, yield higher group homogeneity for both datasets, except for large groups. The reason for this finding is that *P&C* produces too much small groups when constructing large groups with $P = 200$, which increases ICC values. This phenomenon will be observed in the following evaluations.

On the other hand, the standard deviation of constructed groups' sizes can be considered a robustness measure. Both *k*-means and *P&C* methods produce groups in mostly varying sizes, making them inconvenient in most GRS scenarios. However, all the proposed BKM-based schemes ensure maximum group size and construct groups that are very close in their sizes. To demonstrate the proposed schemes' robustness, we present standard deviations of group sizes in Table 14. Note that the smaller the standard deviation, the more robust the constructed groups are.

As can be followed from Table 14, *k*-means and especially *P&C* schemes yield quite large standard deviations, which means the sizes of constructed groups vary in an extended interval. However, RBP_{mul} and RBP_{aug} provide groups with a very close number of members yielding low standard deviations.

5.4.5. *Practical implications of the proposed group identification approaches*

The broad set of experiments with different parameters performed in the previous sections verify that our proposed approaches are more successful than benchmark automatic group identification method when the group recommendations are produced with traditional aggregation techniques. In this section, we perform various additional experiments for providing an overall evaluation of how our approaches are capable of boosting recommendation quality when applied in modern GRSs to

Table 11
*n*DCG results of RBP decorated with DBPs for MLP.

	Group Size (*P*)		Small		Medium		Large	
Aggregation Technique	top-*N*	Method	5	10	25	50	150	200
AVG	5	*P&C*	0.161	0.168	0.189	0.185	0.190	0.186
		RBP	0.176	0.186	0.204	0.220	0.205	0.230
		RBP_{mul}	0.172†	0.191†	0.219†*	0.233†*	0.221†*	0.245†*
		RBP_{aug}	0.177†	0.194†*	0.217†*	0.223†	0.230†*	0.245†*
	10	*P&C*	0.184	0.191	0.202	0.199	0.199	0.209
		RBP	0.187	0.199	0.217	0.225	0.229	0.231
		RBP_{mul}	0.186	0.199†	0.229†*	0.234†*	0.249†*	0.245†*
		RBP_{aug}	0.189	0.199†	0.229†*	0.234†*	0.249†*	0.247†*
	20	*P&C*	0.221	0.222	0.220	0.236	0.248	0.231
		RBP	0.221	0.231	0.253	0.260	0.265	0.275
		RBP_{mul}	0.223	0.232	0.267†*	0.271†*	0.280†*	0.283†
		RBP_{aug}	0.224	0.234	0.260†*	0.268†*	0.282†*	0.285†*
LM	5	*P&C*	0.164	0.164	0.168	0.159	0.149	0.147
		RBP	0.169	0.165	0.158	0.152	0.137	0.154
		RBP_{mul}	0.165	0.170*	0.175*	0.163	0.155*	0.159
		RBP_{aug}	0.165	0.164	0.164	0.167*	0.150	0.163
	10	*P&C*	0.182	0.185	0.187	0.175	0.188	0.166
		RBP	0.180	0.170	0.168	0.159	0.153	0.156
		RBP_{mul}	0.183	0.173	0.173*	0.176*	0.166	0.167
		RBP_{aug}	0.180	0.173	0.176*	0.171	0.163	0.166
	20	*P&C*	0.210	0.209	0.204	0.183	0.165	0.168
		RBP	0.197	0.195	0.181	0.164	0.150	0.156
		RBP_{mul}	0.202	0.199	0.195*	0.187*	0.165*	0.162
		RBP_{aug}	0.205*	0.202*	0.190	0.189*	0.169*	0.165
COP	5	*P&C*	0.169	0.174	0.179	0.192	0.209	0.179
		RBP	0.180	0.185	0.201	0.222	0.230	0.229
		RBP_{mul}	0.179	0.193†*	0.217†*	0.230†	0.243†*	0.246†*
		RBP_{aug}	0.180†	0.195†*	0.208†*	0.223†	0.245†*	0.243†*
	10	*P&C*	0.186	0.194	0.201	0.215	0.205	0.226
		RBP	0.191	0.206	0.225	0.234	0.243	0.249
		RBP_{mul}	0.192	0.208	0.229†	0.246†*	0.259†*	0.256†*
		RBP_{aug}	0.190	0.212*	0.227†	0.245†*	0.262†*	0.258†*
	20	*P&C*	0.225	0.227	0.234	0.240	0.239	0.232
		RBP	0.226	0.237	0.258	0.262	0.275	0.277
		RBP_{mul}	0.228	0.245†*	0.269†*	0.271†*	0.295†*	0.289†*
		RBP_{aug}	0.229	0.248†*	0.269†*	0.264†	0.291†*	0.287†*

† For significance at 95%; w.r.t. *P&C*.
† For significance at 99%.
* For significance at 95%; w.r.t. RBP.

identify groups. To this end, we adopt two recently proposed state-of-the-art frameworks, namely UL (Seo et al., 2018) and IBGR (Nozari et al., 2020), which are explained in detail in Section 2.

In these experiments, we compare the original IBGR and UL methods against their variants in which groups are identified by our BKM approach and its versions empowered with RBP, RBP_{mul} and RBP_{aug}. We consider top-10 group recommendations and all three group formations, i.e., small, medium, and large, by selecting *P* as 10, 50, and 200. Unlike the previous experiments, we also included the ML10M dataset and Precision, Recall, and F1-score metrics to provide a comprehensive examination of how our approaches' performances are affected by dataset sparsity and size. Tables 15–17 present the accuracy outcomes of the experiments conducted for small, medium, and large groups, respectively. We also give the improvement ratios achieved by our approaches in parentheses. Note that since the ML10M does not include any demographic information about users, we cannot perform any experiment to evaluate RBP_{mul} and RBP_{aug} approaches for this dataset.

Based on the obtained accuracy results, it can be concluded that the UL usually outperforms IBGR in terms of all metrics and group formations, and both methods are significantly more efficient than traditional aggregation techniques such as AVG, LM, and COP (see *n*DCG results in the Section 5.4.3). This is because the utilized mechanisms to aggregate individual preferences in both IBGR and UL methods are more advanced than traditional ones. The experimental results also show that empowering UL and IBGR methods with our proposed grouping approaches significantly improves the produced group recommendations' accuracy for all group formations. This finding is more apparent when one of our ultimate RBP_{mul} or RBP_{aug} approaches is employed, which is coherent with the outcomes of the experiments performed in the previous sections. Also, the best accuracy results for large groups are usually achieved by RBP_{mul}, while for small and medium groups by RBP_{aug}. This observation verifies that demographic correlation among users becomes much more crucial in group identification as groups get crowded.

When the results obtained from MLP, MLM, and ML10M datasets are compared, IBGR, UL, and all their variants are usually more successful on the MLP, which is similar to the trend in the *n*DCG results obtained in Section 5.4.3. The main reason for this finding is that user profiles

Table 12
*n*DCG results of RBP decorated with DBPs for MLM.

	Group Size (*P*)		Small		Medium		Large	
Aggregation Technique	top-*N*	Method	5	10	25	50	150	200
AVG	5	*P&C*	0.108	0.113	0.120	0.122	0.153	0.155
		RBP	0.154	0.177	0.205	0.229	0.238	0.237
		RBP_{mul}	0.156†*	0.182†*	0.214†*	0.235†*	0.244†*	0.245†*
		RBP_{aug}	0.155†	0.181†*	0.213†*	0.230†	0.242†*	0.240†
	10	*P&C*	0.123	0.129	0.136	0.143	0.146	0.160
		RBP	0.159	0.176	0.210	0.215	0.243	0.240
		RBP_{mul}	0.159†	0.180†*	0.219†*	0.220†*	0.247†	0.247†*
		RBP_{aug}	0.160†	0.181†*	0.217†*	0.217†	0.245†	0.243†
	20	*P&C*	0.149	0.154	0.158	0.162	0.173	0.173
		RBP	0.173	0.191	0.220	0.231	0.253	0.255
		RBP_{mul}	0.175†*	0.195†*	0.228†*	0.239†*	0.258†*	0.261†*
		RBP_{aug}	0.176†*	0.195†*	0.229†*	0.239†*	0.253†*	0.258†*
LM	5	*P&C*	0.118	0.117	0.116	0.126	0.123	0.123
		RBP	0.141	0.143	0.145	0.137	0.120	0.114
		RBP_{mul}	0.143†*	0.148†*	0.154†*	0.148†*	0.131*	0.128*
		RBP_{aug}	0.144†*	0.149†*	0.155†*	0.143†*	0.132*	0.119
	10	*P&C*	0.123	0.127	0.130	0.124	0.124	0.117
		RBP	0.143	0.145	0.142	0.129	0.112	0.106
		RBP_{mul}	0.145†*	0.148†*	0.150†*	0.139†*	0.124*	0.114*
		RBP_{aug}	0.146†*	0.151†*	0.150†*	0.140†*	0.117	0.114*
	20	*P&C*	0.141	0.141	0.144	0.140	0.128	0.151
		RBP	0.153	0.153	0.150	0.139	0.112	0.103
		RBP_{mul}	0.154†	0.157†*	0.155†*	0.146†*	0.116	0.109*
		RBP_{aug}	0.157†*	0.157†*	0.153†	0.143*	0.115	0.108*
COP	5	*P&C*	0.082	0.080	0.069	0.099	0.083	0.064
		RBP	0.084	0.089	0.078	0.104	0.095	0.071
		RBP_{mul}	0.086*	0.091†*	0.079†*	0.107†*	0.095†	0.075†*
		RBP_{aug}	0.086*	0.092†*	0.079†*	0.107†*	0.094†	0.076†*
	10	*P&C*	0.075	0.076	0.072	0.078	0.080	0.077
		RBP	0.075	0.080	0.075	0.090	0.084	0.083
		RBP_{mul}	0.076	0.082†*	0.078†*	0.096†*	0.088†*	0.087†*
		RBP_{aug}	0.076	0.082†*	0.078†*	0.094†*	0.086†*	0.086†*
	20	*P&C*	0.101	0.098	0.087	0.105	0.102	0.083
		RBP	0.100	0.101	0.091	0.111	0.108	0.093
		RBP_{mul}	0.102*	0.104†*	0.093†*	0.114†*	0.110†*	0.098†*
		RBP_{aug}	0.102*	0.104†*	0.093†*	0.113†*	0.108†	0.098†*

† For significance at 95%; w.r.t. *P&C*.
† For significance at 99%.
* For significance at 95%; w.r.t. RBP.

become much more sparse as the size of the used dataset gets larger, which leads to difficulty in the aggregation process since the number of ratings involved in the aggregation diminishes. On the other hand, the experimental results show that our methods' accuracy improvements are not affected by dataset size and sparsity, and even increase in nearly all schemes, ensuring the proposed approaches' robustness.

In conclusion, all performed experiments show that our proposed user grouping approaches are highly resistant to dataset sparsity or size and can significantly improve any modern group recommendation approach's recommendation quality.

5.5. Insights and discussions

In this study, our main objective is to develop practical grouping approaches to improve the ability of GRS to produce recommendations that satisfy a group of people. We propose employing bisecting *k*-means clustering in place of traditional *k*-means algorithm, which enables producing user groups with a predefined maximum number of members. According to obtained experimental results, BKM grouping approach is beneficial for appropriately assigning users into small- and medium-size groups, especially in large datasets. Also, it both reduces and stabilizes required computation time compared to *k*-means for assigning new users into suitable groups and rearrangement of group structure, which is a great advantage to deal with the curse of dimensionality.

Moreover, experimental studies performed to evaluate the performance of GBP demonstrate that employing PBP and RBP in the phase of detecting groups is more effective than the pure rating vectors of users in terms of the quality of group formation, handling sparse datasets, and reducing computation time. The outcomes also demonstrate that RBP profiling mechanism commonly outperforms the PBP; however, their performance converges as expected as the number of rated items per user increases since the more the number of possessed items, the less its effect on the constructed profile.

Also, the results of conducted experiments for both datasets indicate that *aug* combining method is more effective for small groups as it assumes the individual preferences of users are much more precious than their demographic attributes. On the other hand, the results also present that *mul* combining method provides constructing more homogeneous

Table 13
Computational cost of user grouping methods (in seconds).

		Dataset	
	Method	MLP	MLM
Small Groups ($P = 10$)	*k-means*	0.40	24.11
	P&C	20.64	431.10
	BKM	2.03	235.40
	RBP	0.24	3.82
	RBP_{mul}	1.02	26.31
	RBP_{aug}	1.03	30.77
Medium Groups ($P = 50$)	*k-means*	0.09	4.64
	P&C	20.30	411.60
	BKM	1.94	211.90
	RBP	0.18	3.48
	RBP_{mul}	0.72	25.81
	RBP_{aug}	0.89	26.49
Large Groups ($P = 200$)	*k-means*	0.03	4.58
	P&C	20.20	411.50
	BKM	1.860	205.73
	RBP	0.15	3.35
	RBP_{mul}	0.43	25.55
	RBP_{aug}	0.47	32.35

medium and large groups, which inspires that the demographic structure of groups becomes more critical as the groups get crowded. Also, the outcomes demonstrate that utilizing either of these methods throughout the automatic group identification process is a better choice than employing the benchmark algorithm in terms of providing group recommendations of higher quality.

The accuracy of group recommendations commonly diminishes with increasing group size in extant studies of GRSs (Boratto & Carta, 2011; Boratto et al., 2016), which occurs because these studies focus on individually predicting all existing user ratings and compare them with actual votes to evaluate their proposed approach. Thus, an increase in group size makes it difficult to predict all existing ratings accurately by group recommendation approaches, which leads to diminishing the performance of the system in terms of accuracy. However, in the present study, experiments are conducted for recommending a list of N preferable items for a group instead of providing estimations for each vote of group members. Such preferable items are revealed by some kind of an agreement mechanism among members characterized by the utilized aggregation technique. Therefore, the more people in a group, the more reliable the top-N recommendations since they are obtained relying on opinions of a broader community. Hence, the quality of the produced top-N recommendations enhances with increasing group size, especially when the utilized aggregation technique is either Avg as in (Baltrunas et al., 2010; Kaššák, Kompan, & Bieliková, 2016) or COP as in (Seo et al., 2018). In conclusion, regardless of the number of members in groups, the BKM variants can provide high-quality top-N group recommendations in case the proper aggregation technique is utilized.

Finally, it can be concluded by evaluating overall experimental outcomes that all the proposed grouping approaches are beneficial regardless of utilized aggregation techniques, size of recommendation list, and varying group sizes, which indicates the robustness of the proposed schemes in group recommendation scenarios.

6. Conclusions and future work

Automatic grouping of users before producing recommendations for user groups is a challenging task for group recommender systems since it directly affects the performance of these systems. In this study, we propose a bisecting k-means clustering-based grouping approach that automatically identifies suitable groups of users for group recommendation to improve the overall satisfaction of members. The proposed clustering approach ensures a maximum group size, helps improving accuracy, and alleviating scalability issues. More specifically, the approach applies a bisecting k-means clustering algorithm on the rating vectors of users in order to build a binary decision tree, which then utilized for both constructing groups of users and determining which group a newcomer belongs. The experiments conducted on benchmark datasets for measuring the effectiveness of the proposed approach demonstrate that it outperforms the standard k-means clustering approach in terms of group recommendation quality, as confirmed by statistical significance tests.

We also offer to utilize two different genre-based profiling methods while performing the bisecting k-means clustering to alleviative sparsity and scalability issues related to the similarity calculation process. These approaches aim to map rating vectors of users to genre-based profiles by considering genres of items that are rated by users in order to empower the separation skills of clustering. More specifically, genre-based

Table 14
Standard deviations of group sizes.

		Dataset	
	Method	MLP	MLM
Small Groups ($P = 10$)	k-means	11.44	14.14
	P&C	43.19	147.94
	RBP_{mul}	2.92	2.72
	RBP_{aug}	2.79	2.64
Medium Groups ($P = 50$)	k-means	41.88	56.35
	P&C	106.81	610.43
	RBP_{mul}	12.88	12.45
	RBP_{aug}	11.41	11.22
Large Groups ($P = 200$)	k-means	67.35	192.09
	P&C	496.81	1,371.95
	RBP_{mul}	43.99	51.06
	RBP_{aug}	23.90	57.09

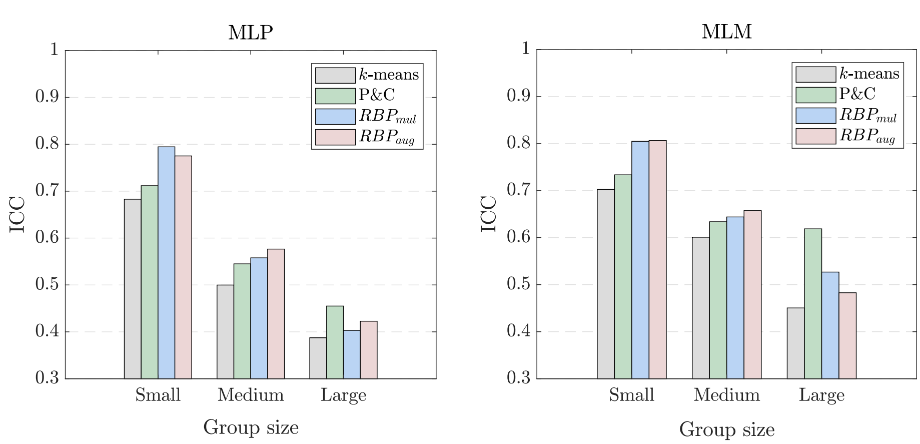

Fig. 6. ICC values for different group formations.

Table 15
Accuracy results for small groups.

Dataset	Method	nDCG@10	Precision@10	Recall@10	F1-Score@10
MLP	*IBGR*	0.381	0.279	0.354	0.261
	BKM	0.409 (7%)	0.296 (6%)	0.380 (7%)	0.283 (8%)
	RBP	0.414† (8%)	0.303† (8%)	0.384† (8%)	0.286† (9%)
	RBP$_{mul}$	0.426† (11%)	0.312† (11%)	0.391† (10%)	0.292† (12%)
	RBP$_{aug}$	0.434† (13%)	0.313† (12%)	0.395† (11%)	0.293† (12%)
	UL	0.466	0.329	0.484	0.328
	BKM	0.498 (6%)	0.349 (5%)	0.516 (6%)	0.351 (6%)
	RBP	0.512† (9%)	0.350 (6%)	0.522 (7%)	0.358† (9%)
	RBP$_{mul}$	0.530† (13%)	0.360† (9%)	0.536† (10%)	0.373† (13%)
	RBP$_{aug}$	0.539† (15%)	0.368† (11%)	0.556† (14%)	0.377† (14%)
MLM	*IBGR*	0.303	0.223	0.203	0.150
	BKM	0.330† (8%)	0.240 (7%)	0.220† (8%)	0.164† (9%)
	RBP	0.339† (11%)	0.249† (11%)	0.227† (11%)	0.166† (10%)
	RBP$_{mul}$	0.342† (12%)	0.253† (13%)	0.241† (18%)	0.170† (13%)
	RBP$_{aug}$	0.343† (13%)	0.252† (13%)	0.244† (19%)	0.170† (13%)
	UL	0.396	0.302	0.391	0.291
	BKM	0.432† (9%)	0.327† (8%)	0.420† (7%)	0.314 (7%)
	RBP	0.446† (12%)	0.329† (9%)	0.431† (10%)	0.320† (9%)
	RBP$_{mul}$	0.467† (17%)	0.333† (10%)	0.435† (11%)	0.330† (13%)
	RBP$_{aug}$	0.467† (17%)	0.335† (11%)	0.447† (14%)	0.333† (14%)
ML10M	*IBGR*	0.231	0.173	0.120	0.103
	BKM	0.255† (10%)	0.189† (9%)	0.135† (12%)	0.116† (12%)
	RBP	0.260† (12%)	0.199† (14%)	0.139† (15%)	0.120† (16%)
	UL	0.344	0.262	0.309	0.254
	BKM	0.386† (12%)	0.295† (12%)	0.350† (13%)	0.290† (14%)
	RBP	0.390† (13%)	0.298† (14)	0.355† (14%)	0.299† (17%)

† For significance at 95%; w.r.t. the state-of-the-art method

Table 16
Accuracy results for medium groups.

Dataset	Method	nDCG@10	Precision@10	Recall@10	F1-Score@10
MLP	*IBGR*	0.291	0.214	0.346	0.200
	BKM	0.315† (8%)	0.232† (8%)	0.359 (3%)	0.210 (5%)
	RBP	0.320† (10%)	0.237† (10%)	0.363 (5%)	0.212 (6%)
	RBP$_{mul}$	0.331† (13%)	0.248† (15%)	0.376† (8%)	0.224† (12%)
	RBP$_{aug}$	0.333† (14%)	0.249† (15%)	0.385† (11%)	0.230† (14%)
	UL	0.424	0.300	0.410	0.265
	BKM	0.442 (4%)	0.317 (5%)	0.421 (3%)	0.273 (3%)
	RBP	0.472† (11%)	0.323 (7%)	0.442 (7%)	0.288 (8%)
	RBP$_{mul}$	0.481† (13%)	0.339† (13%)	0.466† (13%)	0.305† (14%)
	RBP$_{aug}$	0.492† (16%)	0.340† (13%)	0.469† (14%)	0.307† (15%)
MLM	*IBGR*	0.250	0.214	0.244	0.161
	BKM	0.275† (10%)	0.233 (8%)	0.258 (5%)	0.172 (6%)
	RBP	0.280† (11%)	0.240† (12%)	0.261 (6%)	0.176 (9%)
	RBP$_{mul}$	0.284† (13%)	0.248† (16%)	0.273† (11%)	0.184† (14%)
	RBP$_{aug}$	0.285† (14%)	0.249† (16%)	0.273† (11%)	0.185† (14%)
	UL	0.380	0.278	0.319	0.247
	BKM	0.400 (5%)	0.297 (7%)	0.337 (5%)	0.258 (4%)
	RBP	0.409 (7%)	0.310† (11%)	0.319† (11%)	0.274† (10%)
	RBP$_{mul}$	0.434† (14%)	0.317† (14%)	0.364† (14%)	0.284† (14%)
	RBP$_{aug}$	0.434† (14%)	0.315† (13%)	0.366† (14%)	0.285† (15%)
ML10M	*IBGR*	0.205	0.203	0.164	0.141
	BKM	0.228† (11%)	0.222 (9%)	0.180 (9%)	0.153 (8%)
	RBP	0.233† (13%)	0.231† (13%)	0.184† (12%)	0.158† (11%)
	UL	0.318	0.248	0.232	0.213
	BKM	0.343 (7%)	0.271 (9%)	0.249 (7%)	0.227 (6%)
	RBP	0.360† (13%)	0.280† (13%)	0.260† (12%)	0.237† (11%)

† For significance at 95%; w.r.t. the state-of-the-art method.

profiling considers standard features of items and produce a fully dense and lower dimensionality profiles to be employed throughout clustering processes. Also, the dimension of the generated profiles is constant and dependent on the number of available item genres in the system; hence, the required time to compute similarities is reduced and becomes stabilized. Additionally, empirical outcomes demonstrate that utilizing these genre-based profiles contributes to improving satisfaction by group recommendations, as well.

In order to grasp relationships between users firmly in the bisecting *k*-means clustering approach, we also propose to include demography-based correlations into the similarity calculation process in two different strategies, namely multiplicative and augmentative. These strategies estimate ultimate similarities considering demographic correlations between users together with genre-based profile similarities. The multiplicative strategy combines two similarities as equally weighted while the augmentative strategy puts more emphasis on genre-based similarities compared to demographic correlations. The empirical results suggest that utilizing either of these strategies throughout the automatic group identification process significantly enhances the accuracy of the system, where multiplicative strategy is more effective for large groups and augmentative for small and medium ones. Moreover, the empirical outcomes also demonstrate that both strategies are more successful in detecting suitable user groups compared to the state-of-the-art *Predict&Cluster* algorithm. In conclusion, by taking advantage of demographic information of users, the system can construct more homogeneous groups which in turn leads to enhance the overall satisfaction of individual members.

Although the proposed approaches are verified to be effective regardless of the utilized aggregation technique, future research might include exploring the effects of other aggregation techniques, such as borda count, multiplicative, and majority voting together with the proposed user grouping approaches. Also, other clustering techniques, such as fuzzy *c*-means, can be utilized to build the binary decision tree to be utilized for constructing groups. Moreover, rather than building a single tree, a random forest ensemble approach can be examined for exhaustively searching homogeneous groups.

Table 17
Accuracy results for large groups.

Dataset	Method	*nDCG@10*	*Precision@10*	*Recall@10*	*F1-Score@10*
MLP	*IBGR*	0.259	0.200	0.321	0.178
	BKM	0.278 (7%)	0.204 (2%)	0.334 (4%)	0.184 (3%)
	RBP	0.284 (9%)	0.208 (4%)	0.341 (6%)	0.187 (4%)
	RBP_{mul}	0.295† (14%)	0.228† (14%)	0.363† (13%)	0.202† (13%)
	RBP_{aug}	0.293† (13%)	0.217 (8%)	0.352† (9%)	0.199† (11%)
	UL	0.376	0.269	0.384	0.241
	BKM	0.390 (3%)	0.281 (4%)	0.398 (3%)	0.250 (3%)
	RBP	0.410† (9%)	0.289 (7%)	0.411 (6%)	0.260 (7%)
	RBP_{mul}	0.425† (13%)	0.304† (12%)	0.432† (12%)	0.274† (13%)
	RBP_{aug}	0.417† (10%)	0.295† (9%)	0.425† (10%)	0.270† (11%)
MLM	*IBGR*	0.197	0.165	0.211	0.120
	BKM	0.215 (8%)	0.178 (7%)	0.222 (5%)	0.129 (7%)
	RBP	0.221† (12%)	0.184† (10%)	0.234† (10%)	0.134† (11%)
	RBP_{mul}	0.228† (15%)	0.189† (14%)	0.240† (13%)	0.139† (15%)
	RBP_{aug}	0.227† (14%)	0.188† (13%)	0.236† (11%)	0.137† (13%)
	UL	0.375	0.261	0.313	0.233
	BKM	0.390 (4%)	0.278 (6%)	0.326 (4%)	0.242 (4%)
	RBP	0.413† (10%)	0.283 (8%)	0.337 (7%)	0.253 (8%)
	RBP_{mul}	0.431† (14%)	0.297† (13%)	0.354† (13%)	0.266† (14%)
	RBP_{aug}	0.422† (12%)	0.295† (12%)	0.349† (11%)	0.263† (12%)
ML10M	*IBGR*	0.167	0.145	0.173	0.117
	BKM	0.183† (9%)	0.162† (11%)	0.189† (9%)	0.129† (10%)
	RBP	0.192† (14%)	0.169† (16%)	0.194† (12%)	0.136† (15%)
	UL	0.325	0.249	0.279	0.218
	BKM	0.343 (5%)	0.269† (8%)	0.295 (5%)	0.236 (8%)
	RBP	0.363† (11%)	0.279† (12%)	0.310† (11%)	0.243† (11%)

† For significance at 95%; w.r.t. the state-of-the-art method.

CRediT authorship contribution statement

Emre Yalcin: Conceptualization, Investigation, Methodology, Software, Writing - original draft. **Alper Bilge:** Conceptualization, Validation, Methodology, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Álvarez Márquez, J. O., & Ziegler, J. (2016). Hootle+: A group recommender system supporting preference negotiation. In T. Yuizono, H. Ogata, U. Hoppe, & J. Vassileva (Eds.), *Collaboration and Technology* (pp. 151–166). Cham: Springer International Publishing.

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., & Yu, C. (Aug. 2009). Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment, 2*(1), 754–765.

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*(8–9), 687–714.

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. RecSys '10* (pp. 119–126). New York, NY, USA: ACM.

Barzegar Nozari, R., & Koohi, H. (2020). A novel group recommender system based on members' influence and leader impact. *Knowledge-Based Systems, 205*, Article 106296.

Bilge, A. & Polat, H. (2011). An improved profile-based CF scheme with privacy. In: Proceedings - 5th IEEE International Conference on Semantic Computing, ICSC 2011.

Bilge, A., & Polat, H. (2013). A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. *Information Processing & Management, 49*(4), 912–927.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (Oct 2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10), P10008.

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems, 23*(6), 520–528.

Boratto, L., & Carta, S. (2011). *State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups* (pp. 1–20). Berlin, Heidelberg: Springer Berlin Heidelberg.

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. ICEIS 2014. SCITEPRESS - Science and Technology Publications Lda, Setubal, PRT* (pp. 564–572).

Boratto, L., & Carta, S. (2015). Art: Group recommendation approaches for automatically detected groups. *International Journal of Machine Learning and Cybernetics, 6*(6), 953–980.

Boratto, L., & Carta, S. (Oct 2015). The rating prediction task in a group recommender system that automatically detects groups: architectures, algorithms, and performance evaluation. *Journal of Intelligent Information Systems, 45*(2), 221–245.

Boratto, L., Carta, S., Chessa, A., Agelli, M., & Clemente, M. L., (2009). Group recommendation with automatic identification of users communities. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. Vol. 3. pp. 547–550.

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.

Boratto, L., Carta, S., & Fenu, G. (2017). Investigating the role of the rating prediction task in granularity-based group recommender systems and big data scenarios. *Information Sciences, 378*, 424–443.

Boratto, L., Carta, S., Fenu, G., Mulas, F., & Pilloni, P. (2016). Influence of rating prediction on group recommendation's accuracy. *IEEE Intelligent Systems, 31*(6), 22–27.

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

Cantador, I., & Castells, P. (2011). Extracting multilayered communities of interest from semantic user profiles: Application to group modeling and hybrid recommendations. Computers in Human Behavior 27 (4), 1321–1336, social and Humanistic Computing for the Knowledge Society.

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. GROUP '05* (pp. 120–123). New York, NY, USA: ACM.

Choi, K., & Suh, Y. (Jan. 2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems, 37*, 146–153.

Crossen, A., Budzik, J., & Hammond, K. J. (2002). Flytrap: Intelligent group music recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces. IUI '02* (pp. 184–185). New York, NY, USA: ACM.

Delic, A., Neidhardt, J., Nguyen, T. N., & Ricci, F. (June 2018). An observational user study for group recommender systems in the tourism domain. *Information Technology & Tourism, 19*(1), 87–116.

Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of big data - evolution, challenges and research agenda. *International Journal of Information Management, 48*, 63–71.

Esmaeili, L., Mardani, S., Golpayegani, S. A. H., & Madar, Z. Z. (2020). A novel tourism recommender system in the context of social commerce. *Expert Systems with Applications, 149*, Article 113301. URL http://www.sciencedirect.com/science/article/pii/S0957417420301263.

Fatemi, M., & Tokarchuk, L. (2012). An empirical study on imdb and its communities based on the network of co-reviewers. In: Proceedings of the First Workshop on Measurement, Privacy, and Mobility. MPM '12. Association for Computing Machinery, New York, NY, USA.

Fatemi, M., & Tokarchuk, L. (2013). A community based social recommender system for individuals groups. In *2013 International Conference on Social Computing* (pp. 351–356).

Felfernig, A., Polat-Erdeniz, S., Uran, C., Reiterer, S., Atas, M., Tran, T. N., Azzoni, P., Kiraly, C., & Dolui, K. (2019). An overview of recommender systems in the internet of things. *Journal of Intelligent Information Systems, 52*(2), 285–309. https://doi.org/10.1007/s10844-018-0530-7

Felfernig, A., Stettinger, M., Boratto, L., & Tkalcic, M. (2018). Group Recommender Systems: An Introduction. In *Springer Briefs in Electrical and Computer Engineering*. US: Springer.

Feng, S., & Cao, J. (2017). Improving group recommendations via detecting comprehensive correlative information. *Multimedia Tools and Applications, 76*(1), 1355–1377.

Fortunato, S., & Castellano, C. (2012). *Community Structure in Graphs* (pp. 490–512). New York, New York, NY: Springer.

Gorla, J., Lathia, N., Robertson, S., & Wang, J. (2013). Probabilistic group recommendation via information matching. In *Proceedings of the 22Nd International Conference on World Wide Web. WWW '13* (pp. 495–504). New York, NY, USA: ACM.

Ha, T., & Lee, S. (2017). Item-network-based collaborative filtering: A personalized recommendation method based on a user's item network. *Information Processing & Management, 53*(5), 1171–1184. URL: http://www.sciencedirect.com/science/article/pii/S0306457316304794.

Hammou, B. A., Lahcen, A. A., & Mouline, S. (2019). A distributed group recommendation system based on extreme gradient boosting and big data technologies. *Applied Intelligence*, 1–22.

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. SIGIR '99* (pp. 230–237). New York, NY, USA: ACM.

Huang, Z., Xu, X., Zhu, H., & Zhou, M. (2020). An efficient group recommendation model with multiattention-based neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 31*(11), 4461–4474.

Hurtado, R., Bobadilla, J., Gutiérrez, A., & Alonso, S. (2020). A collaborative filtering probabilistic approach for recommendation to large homogeneous and automatically detected groups. *International Journal of Interactive Multimedia & Artificial Intelligence, 6*(2).

Jeong, H. J., & Kim, M. H. (2019). Hggc: A hybrid group recommendation model considering group cohesion. *Expert Systems with Applications, 136*, 73–82.

Kaššák, O., Kompan, M., & Bieliková, M. (2016). Personalized hybrid recommendation for group of users: Top-n multimedia recommender. *Information Processing & Management, 52*(3), 459–477.

Khazaei, E., & Alimohammadi, A. (2018). An automatic user grouping model for a group recommender system in location-based social networks. *ISPRS International Journal of Geo-Information., 7*, 67.

Lancichinetti, A., Fortunato, S., & Kertész, J. (Mar 2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics, 11* (3), Article 033015.

Li, Y., Wang, D., He, H., Jiao, L., & Xue, Y. (2017). Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems. *Neurocomputing, 249*, 48–63. URL: http://www.sciencedirect.com/science/article/pii/S0925231217304836.

Li, Hao, Xia, Huan, Kang, Yan, & Uddin, Mohammad Nashir (2018). Iptv program recommendation based on combination strategies. *MATEC Web of Conferences, 169*, 01003. https://doi.org/10.1051/matecconf/201816901003

Liu, Y., Wang, B., Wu, B., Zeng, X., Shi, J., & Zhang, Y. (2016). Cogrec: A community-oriented group recommendation framework. In W. Che, Q. Han, H. Wang, W. Jing, S. Peng, J. Lin, G. Sun, X. Song, H. Song, & Z. Lu (Eds.), *Social Computing* (pp. 258–271). Singapore: Springer Singapore.

Maraj, A., Shehu, A., Maraj, D., & Sefa, R. (2017). Bandwidth allocation for multiple iptv users sharing the same link: A case study of telecom of kosovo. *Turkish Journal of Electrical Engineering & Computer Sciences, 25*, 3227–3239.

Masthoff, J. (2011). Group recommender systems: Combining individual models. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 677–702). US, Boston, MA: Springer.

Masthoff, J. (2015). *Group Recommender Systems: Aggregation, Satisfaction and Group Attributes* (pp. 743–776). US, Boston, MA: Springer.

McCarthy, J. F., & Anagnost, T. D. (1998). MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (pp. 363–372).

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* (pp. 86–91). AAAI Press.

Newman, M. E. J., & Girvan, M. (Feb 2004). Finding and evaluating community structure in networks. *Physical Review E, 69*(2).

Nguyen, T. N., & Ricci, F. (April 2018). A chat-based group recommender system for tourism. *Information Technology & Tourism, 18*(1), 5–28.

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications, 92*, 507–520. URL http://www.sciencedirect.com/science/article/pii/S0957417417306577.

Nunes, I., & Jannach, D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction, 27*(3–5), 393–444. https://doi.org/10.1007/s11257-017-9195-0

O'Connor, M., Cosley, D., Konstan, J. A., & Riedl, J. (2001). *PolyLens: A Recommender System for Groups of Users* (pp. 199–218). Netherlands, Dordrecht: Springer.

Papadakis, H., Panagiotakis, C., & Fragopoulou, P. (2017). Scor: A synthetic coordinate based recommender system. *Expert Systems with Applications, 79*, 8–19.

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).

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative Filtering Recommender Systems* (pp. 291–324). Berlin, Heidelberg: Springer Berlin Heidelberg.

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.

Shi, J., Wu, B., & Lin, X. (2015). A latent group model for group recommendation. In *2015 IEEE International Conference on Mobile Services* (pp. 233–238).

Sridevi, M., & Rao, R. R. (2017). Decors: A simple and efficient demographic collaborative recommender system for movie recommendation. *Advances in Computational Sciences and Technology, 10*(7), 1969–1979.

Turk, A. M., & Bilge, A. (2019). Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks. *Expert Systems with Applications, 115*, 386–402.

van den Berg, R., Kipf, T.N., & Welling, M. (2017). Graph convolutional matrix completion.

Villavicencio, C., Schiaffino, S., Diaz-Pace, J. A., & Monteserin, A. (2019). Group recommender systems: A multi-agent solution. *Knowledge-Based Systems, 164*, 436–458.

Wang, Q., & Fleury, E. (2011). Uncovering overlapping community structure. In L. da F. Costa, A. Evsukoff, G. Mangioni, & R. Menezes (Eds.), *Complex Networks* (pp. 176–186). Berlin, Heidelberg: Springer Berlin Heidelberg.

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.

Wang, X., Tan, Q., & Goh, M. (2020). Attention-based deep neural network for internet platform group users' dynamic identification and recommendation. *Expert Systems with Applications, 160*, Article 113728.

Yalcin, E., & Bilge, A. (2020). Binary multicriteria collaborative filtering. *Turkish Journal of Electrical Engineering & Computer Sciences, 28*(6), 3419–3437.

Yalcin, E., & Bilge, A. (2020). A personality-based aggregation technique for group recommendation. *Eskisehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering, 21*(4), 486–498.

Yalcin, E., Ismailoglu, F., & Bilge, A. (2021). An entropy empowered hybridized aggregation technique for group recommender systems. *Expert Systems with Applications, 166*, Article 114111. URL: http://www.sciencedirect.com/science/article/pii/S0957417420308617.

Zhao, C., Li, X., & Cang, Y. (2015). Bisecting k-means clustering based face recognition using block-based bag of words model. *Optik – International Journal for Light and Electron Optics, 126*(19), 1761–1766.

Zhao, W.-L., Deng, C.-H., & Ngo, C.-W. (2018). k-means: A revisit. *Neurocomputing, 291*, 195–206.