Expert Systems With Applications 183 (2021) 115396

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Group recommender system based on genre preference focusing on reducing the clustering cost

Young-Duk Seo [a], Young-Gab Kim [b,*], Euijong Lee [c], Hyungjin Kim [d]

[a] Department of Computer Engineering, Inha University, 100 Inha-ro, Michuhol-gu, Incheon 22212, Republic of Korea
[b] Department of Computer and Information Security, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul, Republic of Korea
[c] Department of Computer Science, College of Electrical and Computer Engineering, Chungbuk National University, Cheongju-si 28644, South Korea
[d] Hi-Tech Automation Development II, SK Holdings, SK U-Tower, 9, Seongnam-daero 343beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, Republic of Korea

ARTICLE INFO

Keywords:
Recommender system
Group recommendation
Aggregation method
Clustering
Genre preference

ABSTRACT

The most significant advantage of the group recommender system over personalization is the low computational cost because the former analyzes the preferences of many users at once by integrating their preferences. The clustering step is the most time-consuming part of the entire process in a group recommender system. Existing studies either measured the similarities among all users or utilized a clustering algorithm based on the item preference vector to form the groups. However, these existing clustering methods overlooked the clustering cost, and the time complexity was not significantly better than that for personalized recommendations. Therefore, we propose a group recommender system based on the genre preferences of users to dramatically reduce the clustering cost. First, we define a genre preference vector and cluster the groups using this vector. Our group recommender system can reduce the time complexity more efficiently because the number of genres is significantly smaller than the number of items. In addition, we propose a new item preference along with genre weight to subdivide the preferences of users. The evaluation results show that the genre-based group recommender system significantly improves the time efficiency in terms of clustering. Clustering time was about five times faster when using k-means. In addition, for the Gaussian mixture model (GMM), it was about fifty times faster in MovieLens 100 k and about five hundred times faster in Last.fm. The normalized discounted cumulative gain (NDCG) (i.e., accuracy) is not much different from that of the item-based existing studies and is even higher when the number of users is low in a group in MovieLens 100 k.

1. Introduction

A recommender system is mainly classified into personalized and group recommendations (Seo, 2018). In terms of using the set of users, a group recommendation can be similar to collaborative filtering in a personalized recommendation. However, there is a significant difference in whether the target of recommendation is a group or an individual. Collaborative filtering, which is the most popular approach in the personalized recommendation area, recommends items to each user based on other individuals who are found to have similar preferences (Seo et al., 2017; Jain et al., 2020). On the other hand, group recommendation forms a group consisting of a set of users with similar tendencies. As a consequence, all the users in a group receive the same recommendation results (Seo et al., 2018; Dara et al., 2019).

There are several reasons for conducting group recommendation studies. First, most people perform many actions within a group of people who are intimate with them (Li et al., 2018) because human beings are social by nature (Masthoff, 2015). For example, people prefer to watch movies with their friends or families rather than watching alone. The same holds for going out to dinner at a restaurant or watching a sports game. Furthermore, the group recommender system has an advantage in terms of time efficiency when compared to personalization because calculating a large number of users is a time-consuming task (Boratto et al., 2016).

For group recommendations, it is necessary to form a set of users with similar tendencies. This process is termed group clustering (Borattoet al., 2016), group identification (Seo et al., 2018), or community detection (Boratto et al., 2009). Most studies have formed various groups based on the weighted user similarity network (WUSN) built by using the similarity measures among the users, such as the

* Corresponding author.
E-mail addresses: mysid88@inha.ac.kr (Y.-D. Seo), alwaysgabi@sejong.ac.kr (Y.-G. Kim), kongjjagae@cbnu.ac.kr (E. Lee), flslzk@gmail.com (H. Kim).

https://doi.org/10.1016/j.eswa.2021.115396
Received 10 February 2020; Received in revised form 17 April 2021; Accepted 8 June 2021
Available online 12 June 2021
0957-4174/

Table 1
Classification of existing group recommendation studies.

	Clustering		Group recommender algorithm	
	Input value	Clustering Method	Input value	Aggregation method
MusicFX (McCarthy & Anagnost, 1998)	n/a	n/a	Genre preference	AwM
Guo et al. (2016)	n/a	n/a	Item preference with social factor	Avg, MP, LM
Flytrap (Crossen et al., 2002)	Genre information (not preference value)	SN of genre	Item preference	SC
Guo et al. (2019)	Item preference	Random	Item preference	BC
Baltrunas (2010)	Item preference	WUSN (PCC)	Item preference	Avg, LM, BC
Pujahari & Padmanabhan (2015)	Item preference	WUSN (PCC)	Item preference	AU, MP, LM
Mahyar et al. (2017)	Item preference	WUSN (COS, BS)	Item preference	(Weighted) Avg
Park & Nam (2019)	Item preference	WUSN (PCC, COS)	Item preference	AU
Sacharidis (2019)	Item preference	WUSN (PCC)	Item preference	BC (Pareto optimal), Avg
Kim & Ahn (2008)	Feature of users	GA k-means clustering	n/a	n/a
Boratto et al. (2016)	Item preference	k-means clustering	Item preference	AU, AV, BC, LM, MP
Seo et al. (2018)	Item preference	k-means clustering	Item preference	UL

Pearson correlation coefficient (PCC) and cosine similarity (COS) (Baltrunas et al., 2010; Pujahari & Padmanabhan, 2015; Mahyar et al., 2017; Park & Nam, 2019; Sacharidis, 2019). However, this approach offers no obvious advantage over personalized recommendations because the time complexity of this approach is the same as that of the memory-based approach in collaborative filtering (Bobadilla et al., 2013). Some studies have utilized clustering algorithms such as k-means (Kim & Ahn, 2008; Boratto et al., 2016; Seo et al., 2018) and the Gaussian mixture model (GMM, Shental et al., 2004). They mainly utilized the item preference vectors to form groups. These clustering methods are much more efficient than WUSN, but the time complexity increases in proportion to the number of items.

Clustering is an essential factor and the most time-consuming process in group recommendation. However, most studies have overlooked the time complexity of clustering despite the importance of minimizing it. They mainly utilized item preference vectors to form the groups (Boratto et al., 2016; Seo et al., 2018). Therefore, there is a simple way to reduce the clustering cost: use genre information because the number of genres is significantly smaller than the number of items. For example, in the MovieLens 100 k dataset[1], the number of movies is 1,682, while the number of movie genres is 18. In addition, the number of genres is almost fixed, while the number of items continues to increase. Therefore, it is more beneficial to use genre preference than item preference in the group clustering step.

In this study, we proposed a novel group recommender system using genre information to improve both efficiency and effectiveness. We mainly focused on how to reduce the clustering cost and formed groups based on a genre preference vector. In addition, we subdivided the rating and suggested a new measurement of item preference using genre preference as a weight. Most of the explicit ratings in the recommender system are skewed in positive ratings because the users tend to register positive ratings in general (Bobadilla et al.,2010). Thus, the recommendation results may not change significantly (Hurley & Zhang, 2011). Our granular item preference can guarantee the diversity of the results and improve the performance of a group recommender system. The main contributions of this study are as follows:

Our group recommender system mainly focuses on the clustering step. We utilize the genre preference vector to form groups instead of the item preference vector. Therefore, we can dramatically reduce the clustering cost and increase the efficiency of the group recommender system.

We also proposed a new item preference for users with a weight based on genre preference and utilized it as an input of the recommender system. Consequently, genre preference is the most necessary factor in our group recommender system.

We computed the time taken for clustering to validate the efficiency of our group recommender system based on genre. Furthermore, we measured the accuracy of the recommendation results through the normalized discounted cumulative gain (NDCG). Experimental results indicated that genre preference affected both the efficiency and effectiveness of the group recommender system.

The remainder of this paper is structured as follows. Section 2 analyzes existing studies that focused on clustering. Section 3 explains our group recommender system based on genre preference in detail. In Section 4, we explain the comparative evaluation results. Finally, Section 5 presents a summary of this study as a conclusion and future study directions.

2. Related work

As shown in Table 1, we categorized existing studies into two types: clustering and group recommender algorithms. First, we analyzed whether they utilized genre preference in both steps as input. Then, we reviewed how to form groups in the clustering step. Finally, we investigated which aggregation methods were used as the group recommender algorithms. The aggregation method is the most popular group recommender algorithm in group recommender system studies because of its effectiveness and efficiency (De Pessemier et al., 2014). The aggregation methods include additive utilitarian (AU) (Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Park & Nam, 2019), multiplicative (Mu) (Masthoff, 2015), average (Avg) (Baltrunas et al., 2010; Guo et al., 2016; Mahyar et al., 2017), most pleasure (MP) (Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Guo et al., 2016), simple count (SC) (Crossen et al., 2002), borda count (BC) (Baltrunas et al., 2010; Boratto et al., 2016; Guo et al., 2019; Sacharidis, 2019), average without misery (AwM) (McCarthy & Anagnost, 1998), least misery (LM) (Baltrunas et al., 2010; Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Guo et al., 2016), and upward leveling (UL) (Seo et al., 2018).

Some studies (McCarthy & Anagnost, 1998; Guo et al., 2016) utilized the genre preference or item preference with a weighted factor for the group recommender algorithm. In MusicFX, McCarthy & Anagnost (1998) measured the predicted rating of a group based on genre preference. However, this system regarded all the users in a fitness center as one group instead of dividing the users with similar tastes into multiple groups. Guo et al. (2016) measured the social influence of users in a group based on personal attributes and relationships. The former element contained personality, expertise, and susceptibility, while the latter contained intimacy and preference similarities between users. The researchers created a new rating matrix by multiplying the social influence matrix and rating matrix. Although their method subdivided the rating by adding the social influence to the ratings, they did not suggest a group clustering method such as MusicFX.

[1] https://grouplens.org/datasets/movielens/

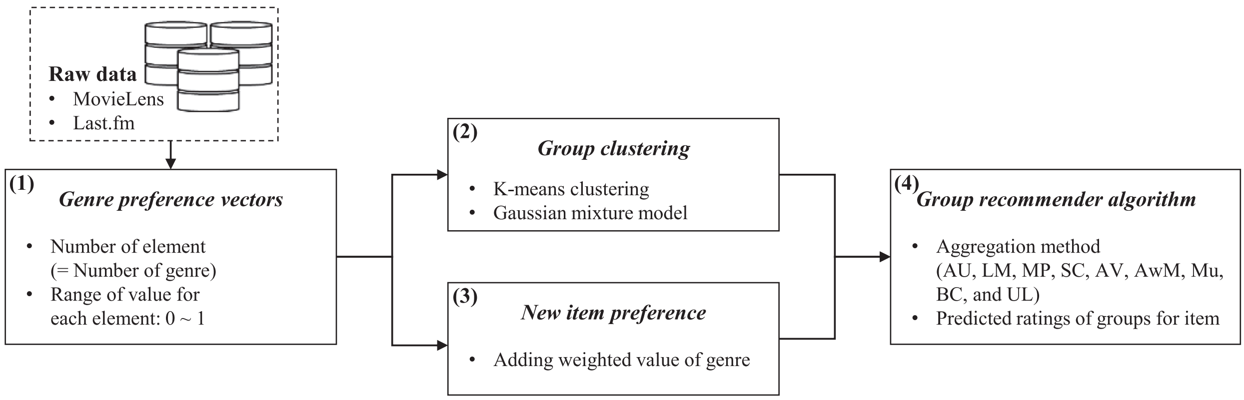


Fig. 1. Framework for proposed group recommendation focusing on genre preference.

Clustering is an essential step in a group recommender system. Most studies (Crossen et al., 2002; Baltrunas et al., 2010; Pujahari & Padmanabhan, 2015; Mahyar et al., 2017; Guo et al., 2019; Park & Nam, 2019; Sacharidis, 2019) constructed a similarity network (SN) to form groups of users or items. Flytrap (Crossen et al., 2002) measured the similarity between genres to construct a network. It calculated the group preference to utilize a voting mechanism that allowed users to vote for their favorite music. However, in Flytrap, Crossen et al. (2002) ignored the clustering step based on user preference. Guo et al. (2019) established a relationship among the items to compare all ratings of a user. Then, they measured new item ratings by utilizing the ELM model. They simply grouped the users randomly and did not consider the clustering process. Baltrunas et al. (2010) proposed the NDCG metric for group recommendation, which was the main goal of the study. They compared all item preference rated by both the target user and his/her neighbor to calculate the similarities between them using PCC. Then, the researchers constructed a WUSN based on similarity to cluster the groups. Furthermore, they evaluated the effectiveness of several aggregation methods based on item preferences. Pujahari & Padmanabhan (2015) measured the similarity by using PCC between items and users to calculate user preferences. They measured the group preference by applying the user preferences as the inputs to the group recommender algorithm, which is a combination of several aggregation methods. However, they could not maximize the advantage of group recommendation because they measured the user preferences based on the personalized recommendation technique. Mahyar et al. (2017) applied the betweenness centrality, which is a group theory, to group recommendations. They first measured the similarity between users based on COS and Bayesian similarity (BS) to calculate the betweenness centrality and constructed a WUSN. Then, they computed the group preferences based on a weighted average. The ratings of highly influential users significantly affected the group preference because this approach utilized betweenness centrality as a weight. Park & Nam (2019) developed a group recommender system for an offline store with a physical presence and not for online services. They applied memory-based collaborative filtering, which is a traditional personalized recommender system method, to group recommendations. They measured the similarities between stores and formed store groups that had similar tendencies. Finally, they determined the recommendation results as a sum of the similarities among stores for an item. Sacharidis (2019) proposed a group recommendation method to present satisfactory recommendation results to a large number of users in a group based on the Pareto optimal. He utilized the rank of the items as a vector and applied it to the Pareto optimal. In addition, he assigned the average of ratings as a weight to each vector. The WUSN might be an accurate method to cluster a group, but there are many disadvantages. First, the cost is high because WUSN compares all relationships between users or items to construct a network. Furthermore, this method is almost the same as memory-based collaborative filtering in personalized recommendation. Therefore, the advantage of the group recommendation disappears when we use WUSN in the clustering step.

We can reduce the overall cost of group recommendation by utilizing a clustering algorithm with regard to WUSN. Thus, some studies (Kim & Ahn, 2008; Boratto et al., 2016; Seo et al., 2018) formed groups using the k-means clustering algorithm. Kim & Ahn (2008) used a genetic algorithm (GA) to optimize the k-means algorithm and defined the new group clustering algorithm as GA k-means clustering. In addition, they proposed an effective market segment method by applying several features of users to the GA k-means clustering. They classified many users into a small number of groups, but they did not provide recommendation results to the groups. Meanwhile, Boratto et al. (2016) and Seo et al. (2018) utilized the k-means clustering algorithm to cluster groups based on item preference vectors. However, they did not use the genre preference in either the group clustering or group algorithm. Furthermore, although there exist more effective clustering algorithms than k-means such as density-based spatial clustering of applications with noise (DBSCAN) and GMM, no studies have applied them to form groups.

As described in existing studies (McCarthy & Anagnost, 1998; Crossen et al., 2002; Kim & Ahn, 2008; Baltrunas et al., 2010; Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Guo et al., 2016, 2019; Mahyar et al., 2017; Seo et al., 2018; Park & Nam, 2019; Sacharidis, 2019), few researchers have considered the genre preference (McCarthy & Anagnost, 1998; Crossen et al., 2002). In other words, almost all studies ignored the usage of genre preference and focused instead on the item preference to cluster a group of users and measure the predicted group preference. Several personalized recommendation studies utilized genre preference as a principal factor in the recommender system (Choi et al., 2012; Qian et al., 2014); however, the influence of genre preference is relatively low in the field of group recommendation.

In contrast to existing studies, we regarded genre preference as the most crucial factor of group recommendation. First, we formed groups based on a genre preference vector to reduce the time complexity and the cost of clustering. We also utilized GMM, which was not used in existing studies, as well as k-means clustering algorithm. Finally, we measured new item preference by using genre preference as a weight.

3. Methodology

This study focuses on genre preference in group recommendations. We proposed a framework for a group recommender system to achieve our goal, as shown in Fig. 1.

We utilized MovieLens and Last.fm2, which are representative datasets in the movie and music domains. First, we measured the average preferences of each genre according to the item preferences of users, and generated genre preference vectors for all users. In the second step, we utilized the k-means clustering algorithm (Hartigan & Wong,

Table 2
Notations used in this paper.

	Notation	Description
Common symbol	N	Number of genres
	u	A user
	g	A group
	i	An item
	θ	Threshold
Genre preference vector	$r_{u,i}$	An item preference of u
	Gen	Set of genres
	i_{gen}	Item-genre vector
	I_u	Set of items rated by u
	r_{u,gen_k}	Genre preference
	$r_{u,gen}$	Genre preference vector
Fine-grained item preference based on genre weight	$w_{u,gen}$	Genre weight
	$\widehat{w}_{u,gen}$	Normalized genre weight
	$p_{u,i}$	Fine-grained item preference
Group recommender algorithm	$p_{g,i}$	Group preference
	$p^{AU}_{g,i}$	Group preference based on AU
	$p^{Mu}_{g,i}$	Group preference based on Mu
	$p^{LM}_{g,i}$	Group preference based on LM
	$p^{MP}_{g,i}$	Group preference based on MP
	$p^{SC}_{g,i}$	Group preference based on SC
	$c^{SC}_{u,i}$	A variable that determines whether $r_{u,i}$ is null or not
	$p^{AV}_{g,i}$	Group preference based on AV
	$c^{AV}_{u,i}$	A variable that determines whether $r_{u,i}$ exceeds a certain threshold (θ) or not
	$p^{BC}_{g,i}$	Group preference based on BC
	$rank_{u,i}$	Ranking score of i for u
	$p^{Dev}_{g,i}$	Group preference based on deviation (Dev)
	$p^{Avg}_{g,i}$	Group preference based on Avg
	$p^{UL}_{g,i}$	Group preference based on UL

1979) and GMM (Shental et al., 2004) to cluster the groups of users based on the genre preference vectors. In the third step, we measured the weighted value of the genre. We also proposed a new item preference for users by adding the weighted value to the rating. Finally, we applied the new item preference to several existing aggregation methods to estimate the group preference.

4. Notations

Table 2 summarizes the notations used in this study. We classified them by subchapter.In this study, we utilized two datasets, MovieLens and Last.fm. Therefore, certain notations were used differently for each dataset. Movielens has an explicit rating ranging from 1 to 5, so we used $r_{u,i}$ as is. We set θ to 4 because many studies regarded a rating of 4 or higher as a positive. In addition, we calculated $\widehat{w}_{u,gen}$ based on min–max normalization. The maximum and minimum values were 5 and 1, respectively. On the other hand, Last.fm has an implicit rating scale. The rating is the number of times a user has listened to a specific artist's song and ranges from 1 to 352,698. Because of its wide range, we used $r_{u,i}$ as $\ln(r_{u,i}+1)$ in Last.fm. We set θ as the average rating of each user and utilized the max and min values of each user in the min–max normalization.

4.1. Group clustering based on genre preference vector

We utilized the genre preference of users in a group instead of the item preference to cluster the groups, because the number of genres is significantly lower than the number of items. For example, in the MovieLens 100 k data, the number of items is approximately 93 times that of the number of genres. Therefore, it is efficient (i.e., computation cost) to cluster groups according to genre preferences rather than item preferences.

A genre is a vital element in this study, so we defined various values related to it before clustering the group. First, we defined the set of genres as Gen. If N genres are included in the recommender system and each genre represents $gen_k (where k = 1, \cdots, N)$, then Gen is defined as follows:

$$Gen = \{\ gen_1,\quad gen_2,\quad \cdots,\quad gen_N\ \}$$

To calculate the genre preference value, we had to determine the genres that are included in item i and rated by user u. Therefore, we defined an item-genre vector i_{gen}, which indicates whether item i belongs to gen_k as follows:

$$i_{gen_k} = \begin{cases} 1 & \text{if } i \text{ is belong to } gen_k \\ 0 & else \end{cases}$$

Then, item-genre vector i_{gen} can be defined as follows:

$$i_{gen} = (\ i_{gen_1},\quad i_{gen_2},\quad \cdots,\quad i_{gen_N}\ )$$

The genre preference is calculated as the average of all ratings for i related to gen_k by using item-genre vectors and item preference values. Then, the genre preference of user i for gen_k is defined as per Eq. (1):

$$r_{u,gen_k} = \frac{\sum_{i \in I_u} r_{u,i} \bullet i_{gen_k}}{\sum_{i \in I_u} i_{gen_k}} \tag{1}$$

Then, we generated the genre preference vector $r_{u,gen}$ with N components based on the measured genre preference values. This is defined as follows:

$$r_{u,gen} = (\ r_{u,gen_1},\quad r_{u,gen_2},\quad \cdots,\quad r_{u,gen_N}\ )$$

In this study, we clustered the groups of all users using k-means (Hartigan & Wong, 1979; ; Arthur & Vassilvitskii, 2007) and GMM clustering algorithms (Shental et al., 2004). Clustering algorithms are much better in terms of efficiency (i.e., time complexity) because they can detect the group automatically and quickly when compared to the WUSN methods (Seo et al., 2018). K-means can be applied to almost all types of data (Raykov et al., 2016) and is the most popular clustering algorithm in the recommender system (Amatriain et al., 2011; Hafshejani et al., 2018). GMM is more flexible than k-means, and often produces better clustering results (Shental et al., 2004). However, existing group recommender systems do not use GMM. Thus, we utilized GMM as well as k-means to derive better recommendation results for users in a group.

4.2. Fine-grained item preference based on genre weight

$r_{u,i}$ was generally obtained by explicit scale ratings (i.e., ranging from 1 to 5) or binary ratings (i.e., positive or negative) in most recommender systems. In such situations, users did not have a wide range of choice for ratings. In addition, they tend to provide a rating when their preference for items is positive (Bobadilla et al., 2010). As a result, the distribution of ratings in the recommender system was often skewed to positive ratings, such as 4 and 5 in scale ratings or positive in binary ratings. In other words, many items rated by users had the same rating.

If a user u provides the same rating to two items i_1 and i_2, this may not mean that the user's degree of preference for them is the same. The existing rating scale systems have difficulty when measuring the fine-grained degrees of item preferences. Although Zhao et al. (2017) considered the closeness of the relationship between users and items as the weight of the rating, we require a method for fine-grained measurement of item preferences.

Most studies have used item preference as the input value not only to calculate the predicted preference value of users for items but also to measure the evaluation metric in recommender systems. This includes root mean squared error (RMSE), mean absolute error (MAE), and NDCG. Generally, recommendation results are provided to users through predicted preference values. The effectiveness of the recommender system was validated through evaluation metrics. If we use a new fine-grained item preference of users as an input value, then we can provide accurate recommendation results to users when compared to using the existing preference values. In addition, this can estimate more accurate effectiveness by using it as the input of the evaluation metric.

In this study, we assumed that the degree of item preferences for users could vary depending on their genre preferences. Therefore, we added the genre weight of users to the rating to measure the fine-grained item preference. The genre weight of a user u is computed by the dot product of two vectors, i_{gen} and $r_{u,gen}$. This can be defined as per Eq. (2).

$$w_{u,gen} = \frac{i_{gen} \bullet r_{u,gen}}{\sum_{k \in N} i_{gen_k}} \tag{2}$$

A higher genre weight can impact existing ratings. For example, we can judge that an item with a 4-point is lower in the preference of users when compared to a 5-point. However, if the genre weight has a great effect on the preference value, a 4-point item may have a higher preference than a 5-point item. To avoid this circumstance, we normalized $w_{u,gen}$ in the range of 0 to 1. Thus, the new item preference $p_{u,i}$ that includes the genre weight can be defined as per Eq. (3).

$$p_{u,i} = \widehat{w}_{u,gen} + r_{u,i} \tag{3}$$

4.3. Group recommendation based on aggregation method

For group recommendation, most studies have used the aggregation method to calculate the predicted group preference. The main objective of this method is to provide recommendation results that satisfy the maximum possible number of group users. In this section, we explain how to calculate $p_{g,i}$ for popular aggregation methods such as AU, Mu, LM, MP, SC, AV, AwM, BC, and UL. We utilized them as a group recommender algorithm in this study.

AU (Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Park & Nam, 2019) and Mu (Masthoff, 2015) are methods that add and multiply the preferences of all users in a group, respectively, to calculate $p_{g,i}$ as defined in Eq. (4) and Eq. (5).

$$p^{AU}_{g,i} = \sum_{u \in g} p_{u,i} \tag{4}$$

$$p^{Mu}_{g,i} = \prod_{u \in g} p_{u,i} \tag{5}$$

The lowest and highest ratings in a group are regarded as $p_{g,i}$ in the LM (Baltrunas et al., 2010; Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Guo et al., 2016) and MP (Pujahari & Padmanabhan, 2015; Boratto et al., 2016; Guoet al., 2016), respectively as defined in Eq. (6) and Eq. (7).

$$p^{LM}_{g,i} = \min_{u \in g} p_{u,i} \tag{6}$$

$$p^{MP}_{g,i} = \max_{u \in g} p_{u,i} \tag{7}$$

SC (Crossen et al., 2002) and AV (Boratto et al., 2016) are counting methods. The SC method counts all ratings as defined in Eq. (8), and the AV method counts all positive ratings as defined in Eq. (9) (i.e., all ratings above a certain threshold θ).

$$\begin{gathered} c^{SC}_{u,i} = \begin{cases} 1 & if\, r_{u,i} \neq null \\ 0 & else \end{cases} \\ p^{SC}_{g,i} = \sum_{u \in g} c_{u,i} \end{gathered} \tag{8}$$

Table 3
Example of BC calculation. All elements in this table represent "$r_{u,i}$ ($rank_{u,i}$)."

	i_1	i_2	i_3	i_4	i_5
u_1	5 (4)	5 (4)	4 (2)	4 (2)	2 (1)
u_2	2 (1)	4 (3)	3 (2)	5 (4)	5 (4)
u_3	4 (3)	4 (3)	2 (2)	1 (1)	5 (5)
$p^{BC}_{g,i}$	**8**	**10**	**6**	**7**	**10**

Table 4
Experimental datasets.

	Domain	#users	#items	#genres	Type of ratings
MovieLens 100 k	Movie	943	1,682	18	Scale rating (ranging from 1 to 5)
MovieLens 25 M	Movie	1,625	62,423	19	Scale rating (ranging from 1 to 5)
Last.fm	Music	1,889	10,663	174	Implicit rating (Song heard)

$$\begin{gathered} c^{AV}_{u,i} = \begin{cases} 1 & if\, r_{u,i} \geq \theta \\ 0 & else \end{cases} \\ p^{AV}_{g,i} = \sum_{u \in g} c_{u,i} \end{gathered} \tag{9}$$

AwM (McCarthy & Anagnost, 1998) is a special Avg method. If there is a negative rating in the ratings of all group users, then $p_{g,i}$ is regarded as zero. The AwM method is defined as per Eq. (10).

$$p^{AwM}_{g,i} = \begin{cases} p^{Avg}_{g,i} & if\, all\, r_{u,i} \geq \theta \\ 0 & else \end{cases} \tag{10}$$

BC (Baltrunas et al., 2010; Boratto et al., 2016; Guo et al., 2019; Sacharidis, 2019) is based on the ranking priority and calculates $p_{g,i}$ to provide a high score to a high-ranking item. Table 3 shows an example of the BC calculation.

$p^{BC}_{g,i}$ is the sum of the rank priority rating as per Eq. (11).

$$p^{BC}_{g,i} = \sum_{u \in g} rank_{u,i} \tag{11}$$

UL (Seo et al., 2018) is the most recent aggregation method. It considers the deviation as the most important factor and is calculated by Eq. (12). α, β, and γ are the weights that represent the relative importance of $p^{Avg}_{g,i}$, $p^{AV}_{g,i}$, and $p^{Dev}_{g,i}$. UL can satisfy many group users when compared to other aggregation methods.

$$\begin{gathered} p^{Dev}_{g,i} = 1 - \frac{\sum_{u \in g} \left(p^{Avg}_{g,i} - r_{u,i}\right)^2}{N} \\ p^{UL}_{g,i} = \alpha p^{Avg}_{g,i} + \beta p^{AV}_{g,i} + \gamma p^{Dev}_{g,i} \\ (\alpha + \beta + \gamma = 1) \end{gathered} \tag{12}$$

5. Experiment and evaluation

In this section, we verified the superiority of the proposed group recommender system by focusing on the genre preference when compared to those focusing on item preference. First, we compared the efficiency by measuring the time taken for group clustering. In addition, NDCG was used as the evaluation metric to validate the effectiveness of the proposed group recommender system.

5.1. Dataset

In this study, we validated the superiority of the proposed group

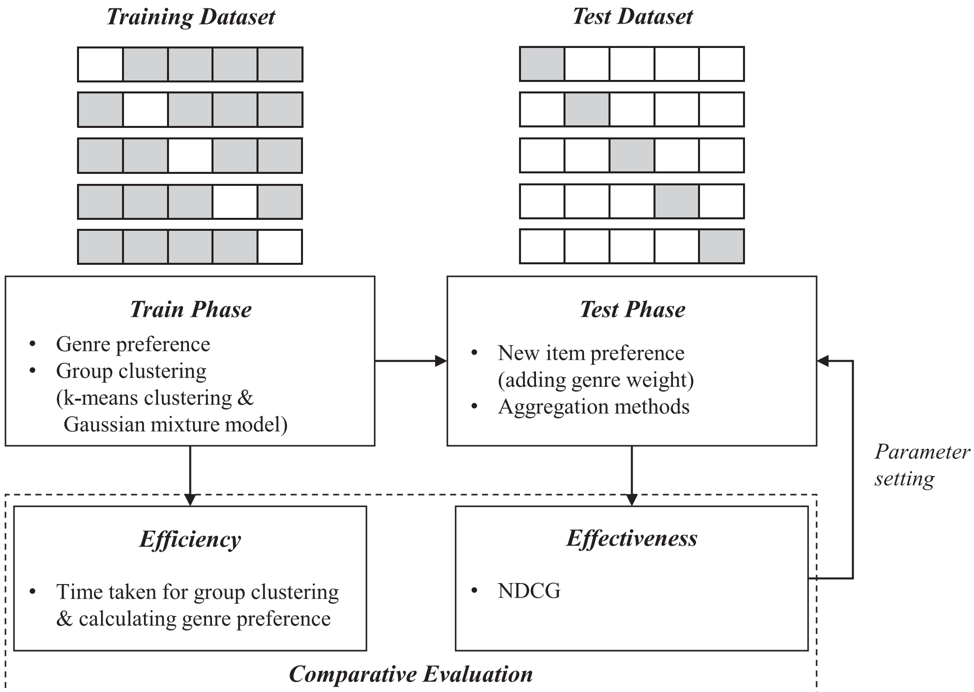


Fig. 2. Comparative evaluation process to measure efficiency and effectiveness.

recommender system by using the MovieLens 100 k, MovieLens 25 M, and Last.fm datasets. They are the most well-known datasets and were widely used in the latest recommender system studies (Seo et al., 2018; Nozari & Koohi, 2020; Wang et al., 2020).

Table 4 shows a detailed description of the experimental datasets. We considered the genre data as an essential factor in this study. The number of genres in MovieLens 100 k is 18 (i.e., action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film noir, horror, musical, mystery, romance, sci-fi, thriller, war, and western), and IMAX genre is added in MovieLens 25 M. In MovieLens 25 M, it has a too large matrix size, so the density of ratings is too low, about 0.24%. On the other hand, MovieLens 100 k is about 6.3%. Therefore, we used 1% of users randomly sampled in MovieLens 25 M instead of using all of them. The Last.fm dataset provides a genre as a tag. In this study, we used only popular tags in the Last.fm that were selected for more than 100 users because there were too many overlapping tags.

5.2. Experimental setup

In this study, we conducted fivefold cross-validation to evaluate the proposed group recommender system focusing on the genre preference. Fig. 2 shows a comparative experiment that aims to measure both efficiency and effectiveness.

In the training phase, we generated the genre preference vectors and clustered a group of users using the training data. Furthermore, to measure the efficiency, we compared the time taken for group clustering by using the group vectors with that by using the item vectors. In the test phase, we first calculated a new item preference based on the genre weight of users, and applied it to various aggregation methods, such as AU, Mu, LM, MP, SC, AV, AwM, BC, and UL, to measure the predicted group preference. Finally, we computed the effectiveness of our group recommender system by using the NDCG. In MovieLens 25 M, unlike other data sets, we selected 1% of users randomly, so we performed the cross-validations 5 times and measured the average NDCG values.

In this study, we utilized k-means and GMM to form the groups. Therefore, we examined baseline studies that used the clustering algorithm (Boratto et al., 2016; Seo et al., 2018) and not WUSN (Baltrunas, 2010; Pujahari & Padmanabhan, 2015; Mahyar et al., 2017; Park & Nam, 2019; Sacharidis, 2019). We compared the two cases: one where the groups were clustered by the genre preference vectors and another by the item preference vectors. The former is the proposed method, and the latter is from existing baseline studies (Boratto et al., 2016; Seo et al., 2018). In addition, we analyzed the change in effectiveness when adding the genre weight to the existing item preference.

Table 5
Hardware environment to measure efficiency.

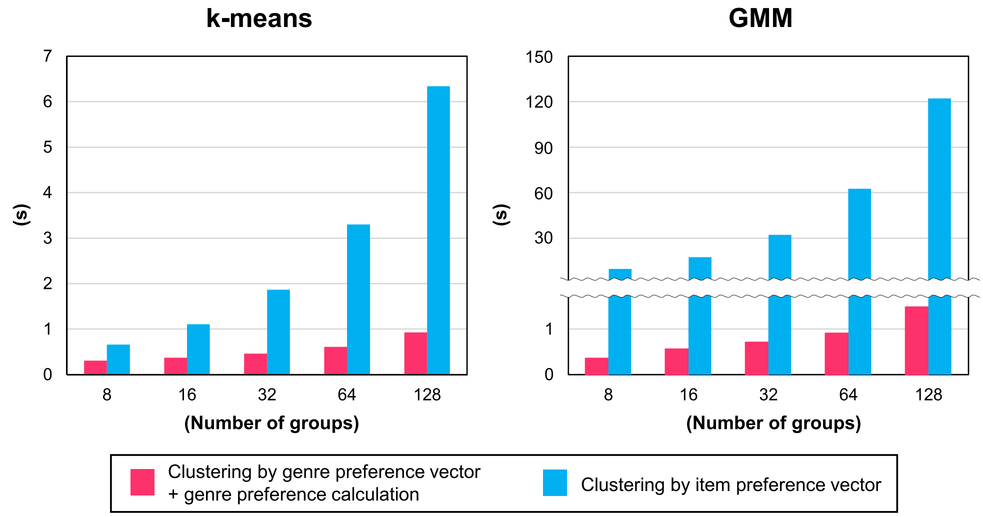
CPU			GPU	Ram	OS
Model	Clock	#Cores			
Intel Xeon E5-2620 v4	2.1 GHz	16	NVIDIA GeForce RTX 2080Ti	16 GB	Window 10 Pro

5.3. Experimental results

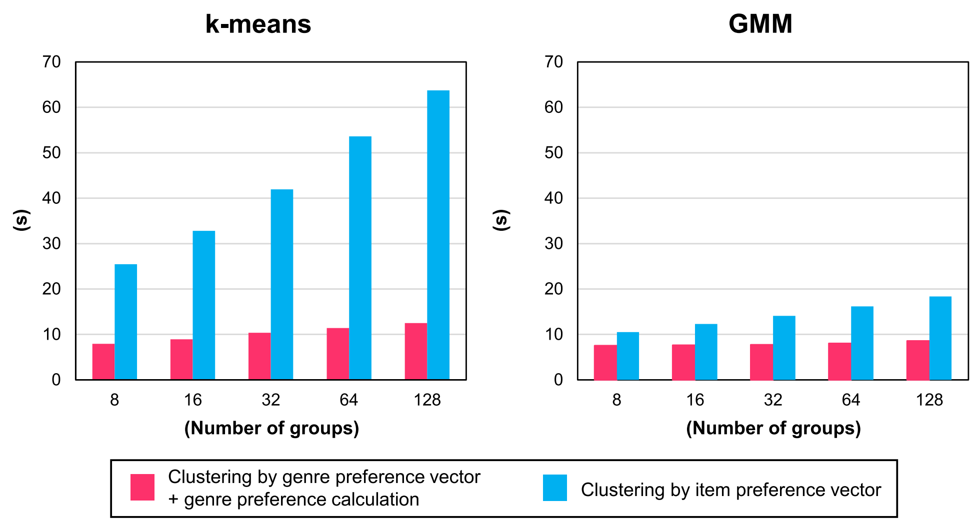
First, we evaluated the efficiency of measuring the time taken for group clustering while using the genre preference vector or item preference vector according to the number of groups (k). We implemented a prototype of group recommendation using Python 3.6 to evaluate the efficiency based on the scikit-learn library[7]. Table 5 lists the specifications of the hardware environment.

The factors that affected the time complexity in the k-means and GMM clustering algorithms were k, the number of vectors (v), and the dimension of the vector (d) (Hartigan & Wong, 1979). v is the same as the number of users and does not change. Therefore, the execution time for group clustering varies depending on k and d.

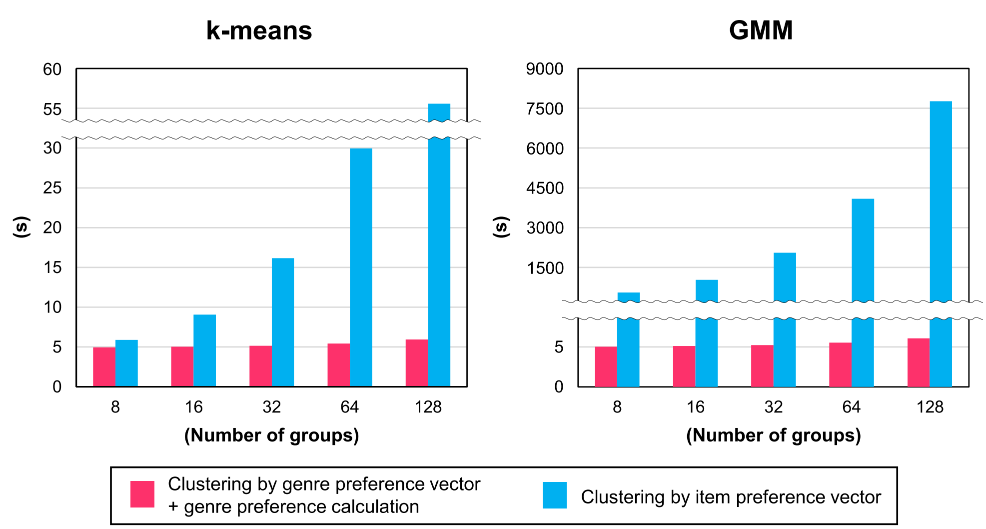
As shown in Fig. 3, it is evident that the execution time increases by approximately two times when the number of groups doubles for cases that use the genre and item preference vectors. Furthermore, the time required for group clustering using the genre preference vector is significantly lower than that for the item preference vector because the number of items is much higher than the number of genres. Clustering time was about two to ten times faster when using k-means in all datasets. In addition, for GMM, it was about twenty to eighty times faster



(a) MovieLens 100k (Genre preference calculation time = 0.2011)



(b) MovieLens 25M (Genre preference calculation time = 7.2770)



(c) Last.fm (Genre preference calculation time = 4.7835)

Fig. 3. Comparison of efficiency in measuring time taken for group clustering.

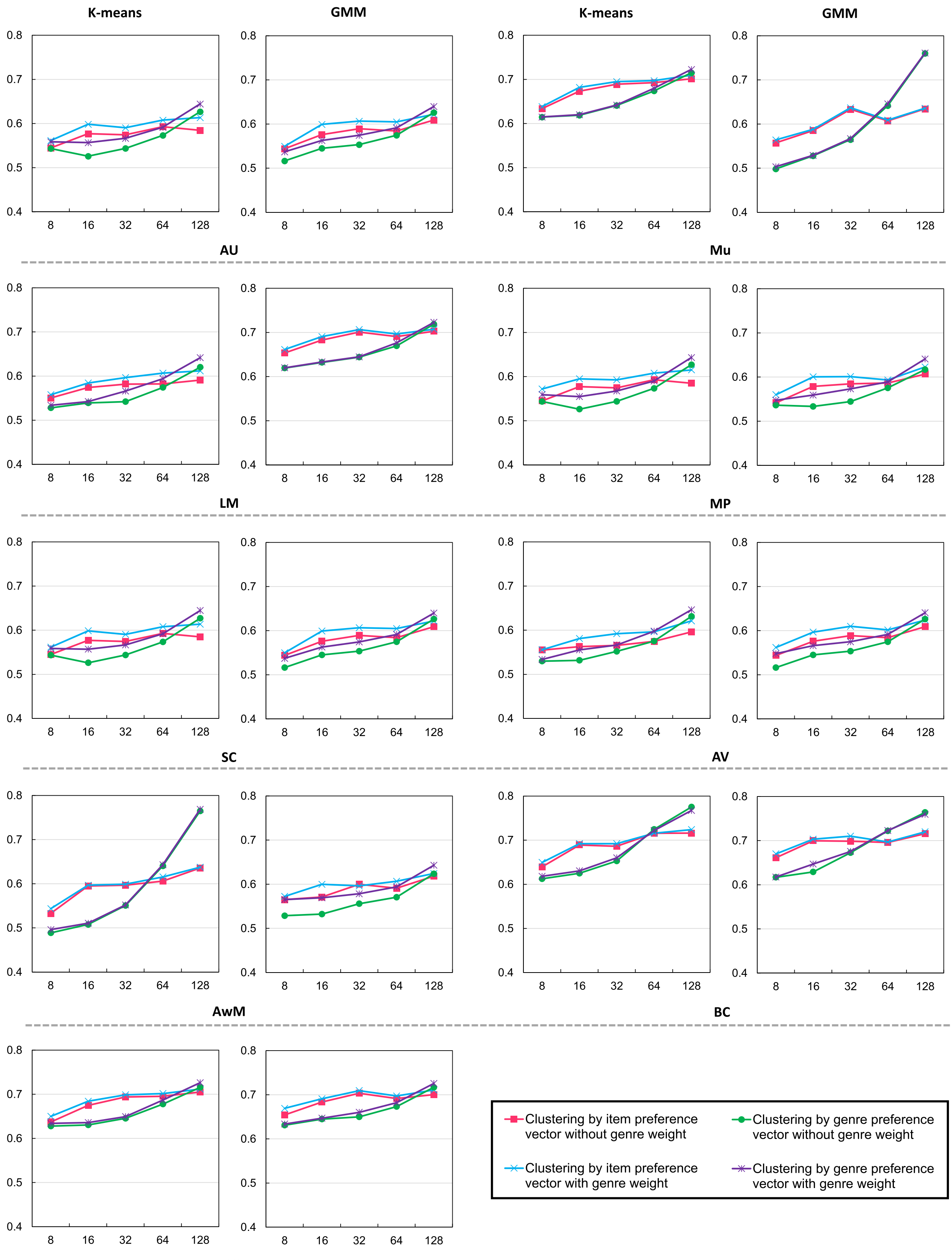


Fig. 4a. Comparative evaluation to measure effectiveness based on MovieLens 100 k (*n* corresponds to 10, x-axis and y-axis are NDCG and number of groups).

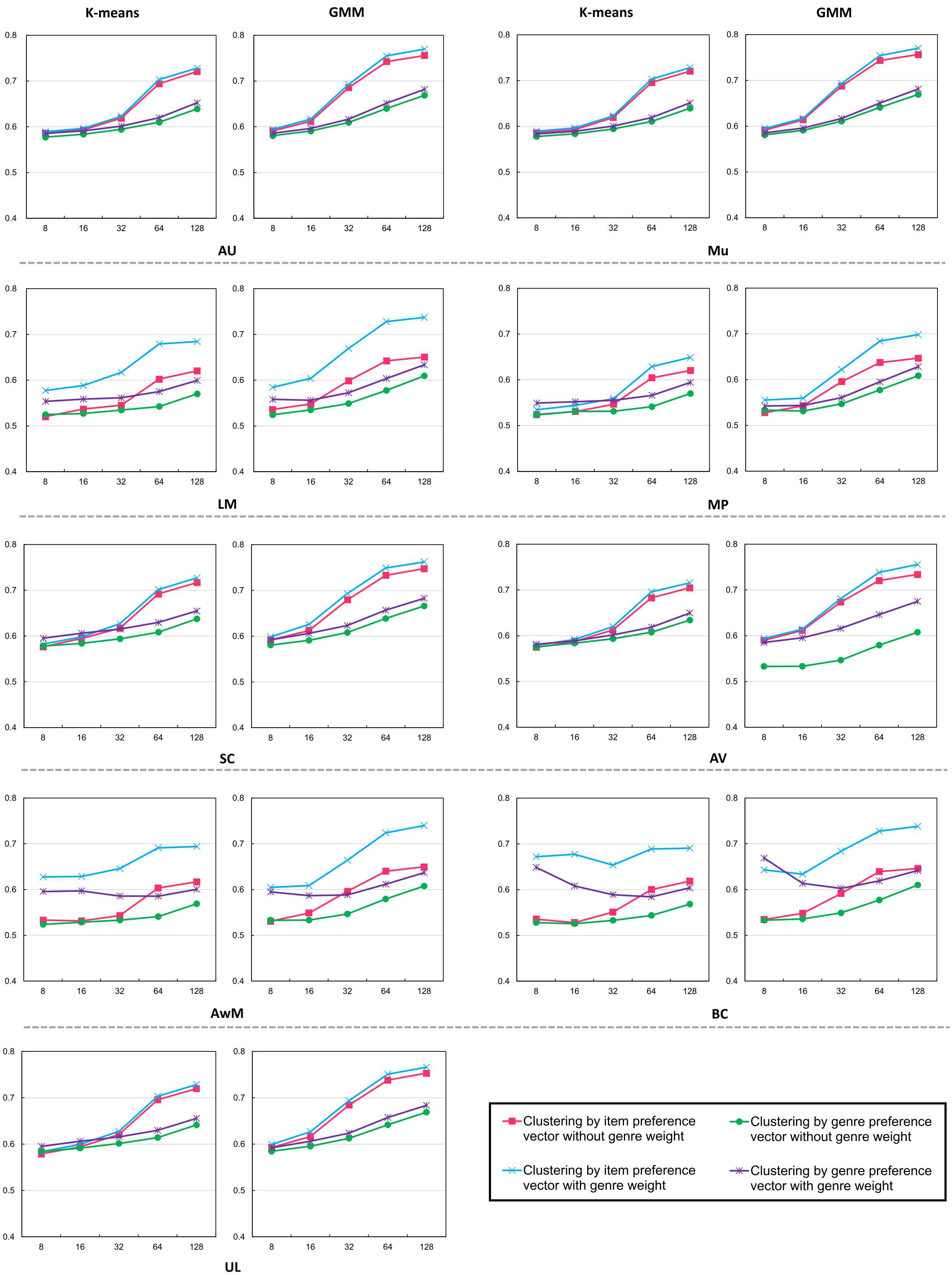


Fig. 4b. Comparative evaluation to measure effectiveness based on MovieLens 25 M (n corresponds to 10, x-axis and y-axis are NDCG and number of groups).

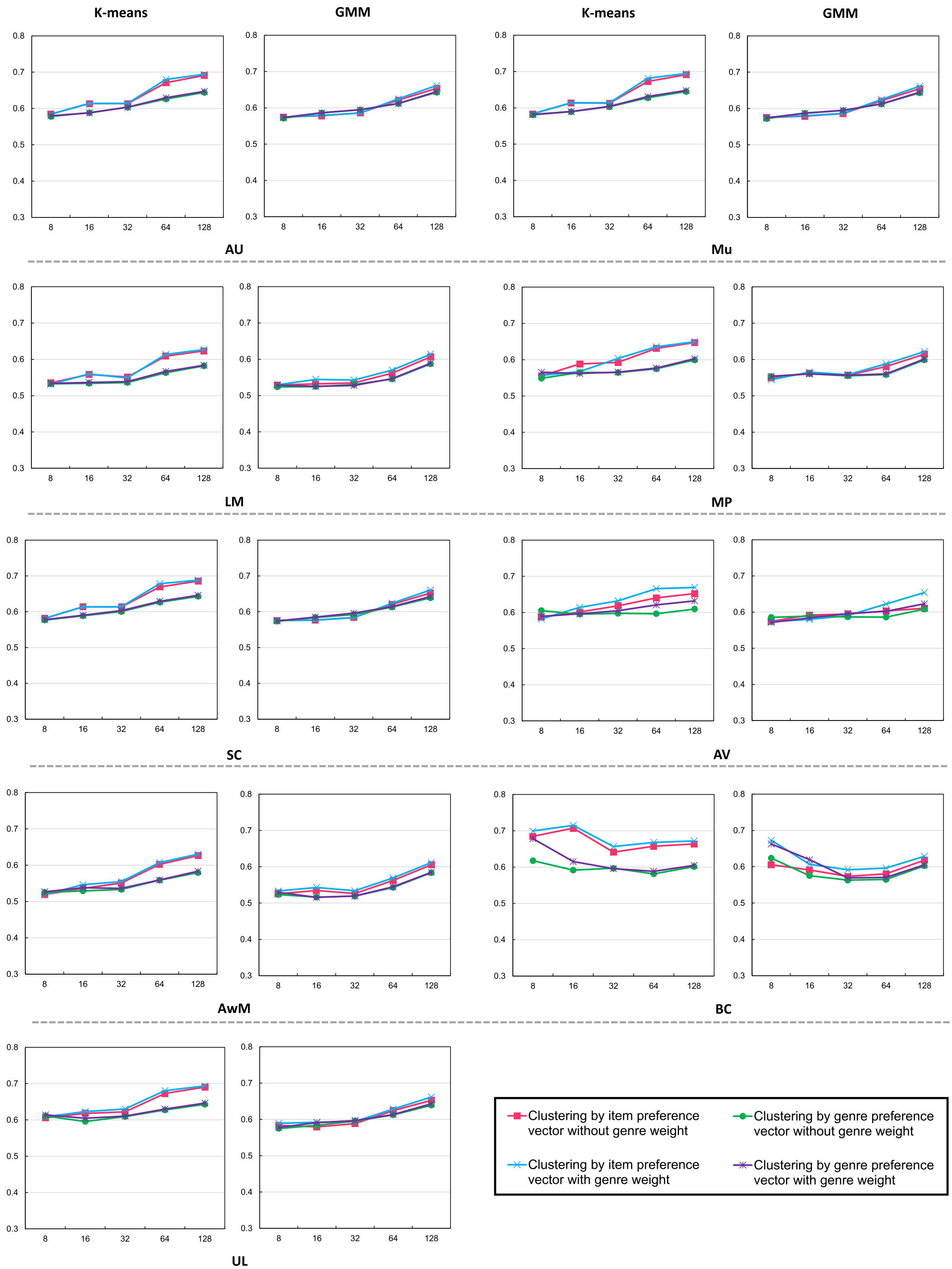


Fig. 4c. Comparative evaluation to measure effectiveness based on Last.fm (*n* corresponds to 10, x-axis and y-axis are NDCG and number of groups).

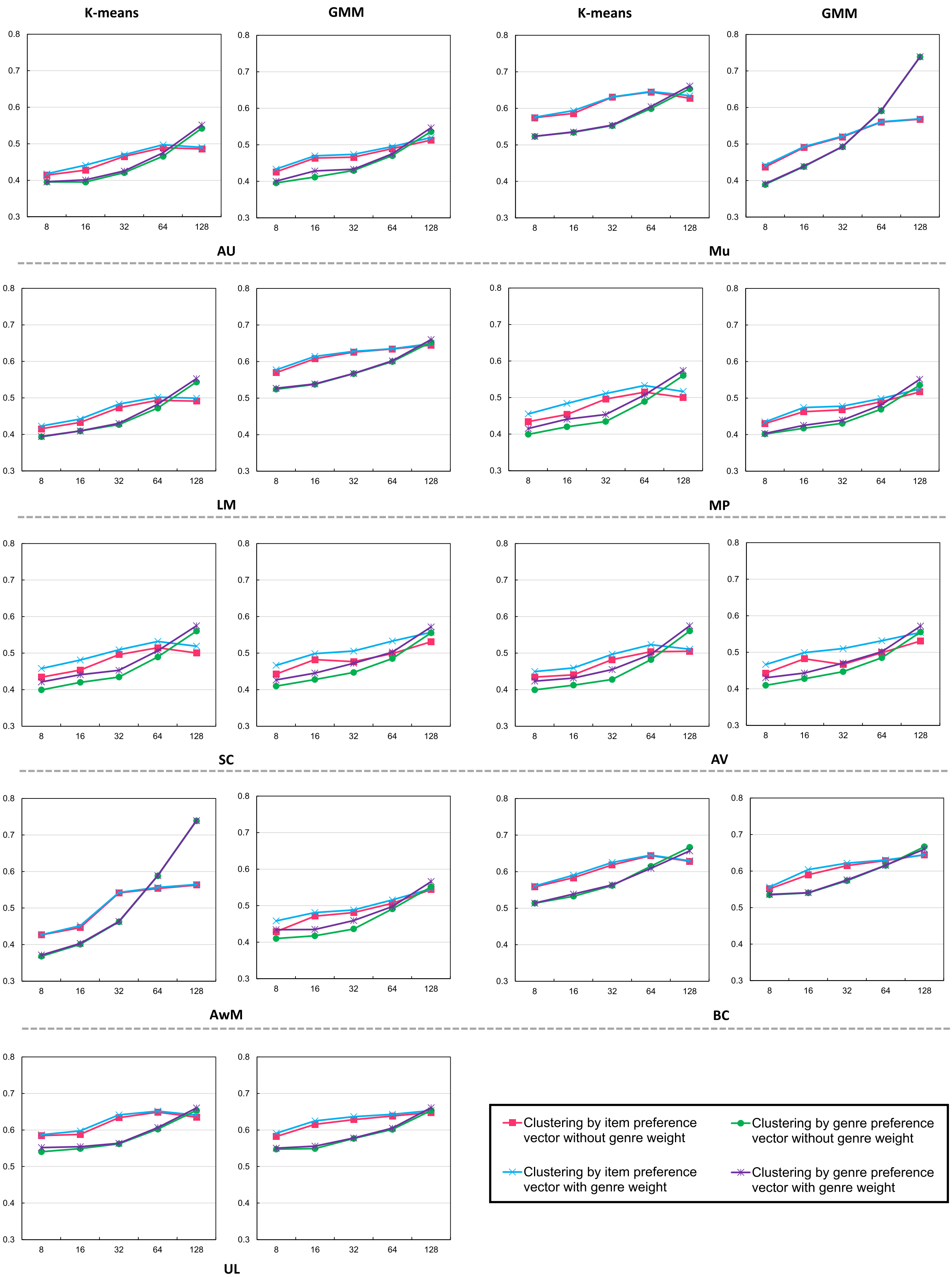


Fig. 5a. Comparative evaluation to measure effectiveness based on MovieLens 100 k (n corresponds to 20, x-axis and y-axis are NDCG and number of groups).

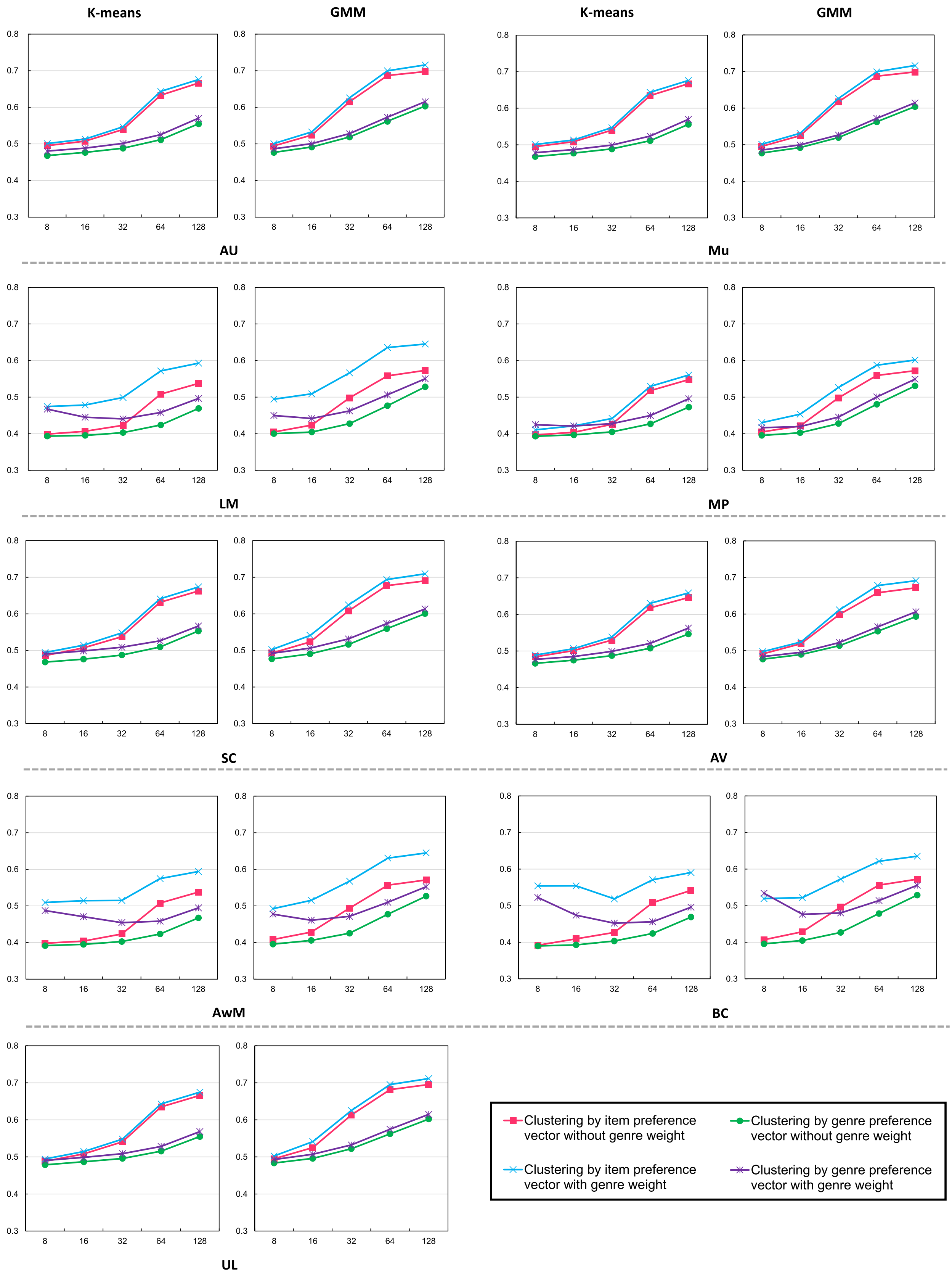


Fig. 5b. Comparative evaluation to measure effectiveness based on MovieLens 25 M(n corresponds to 20, x-axis and y-axis are NDCG and number of groups).

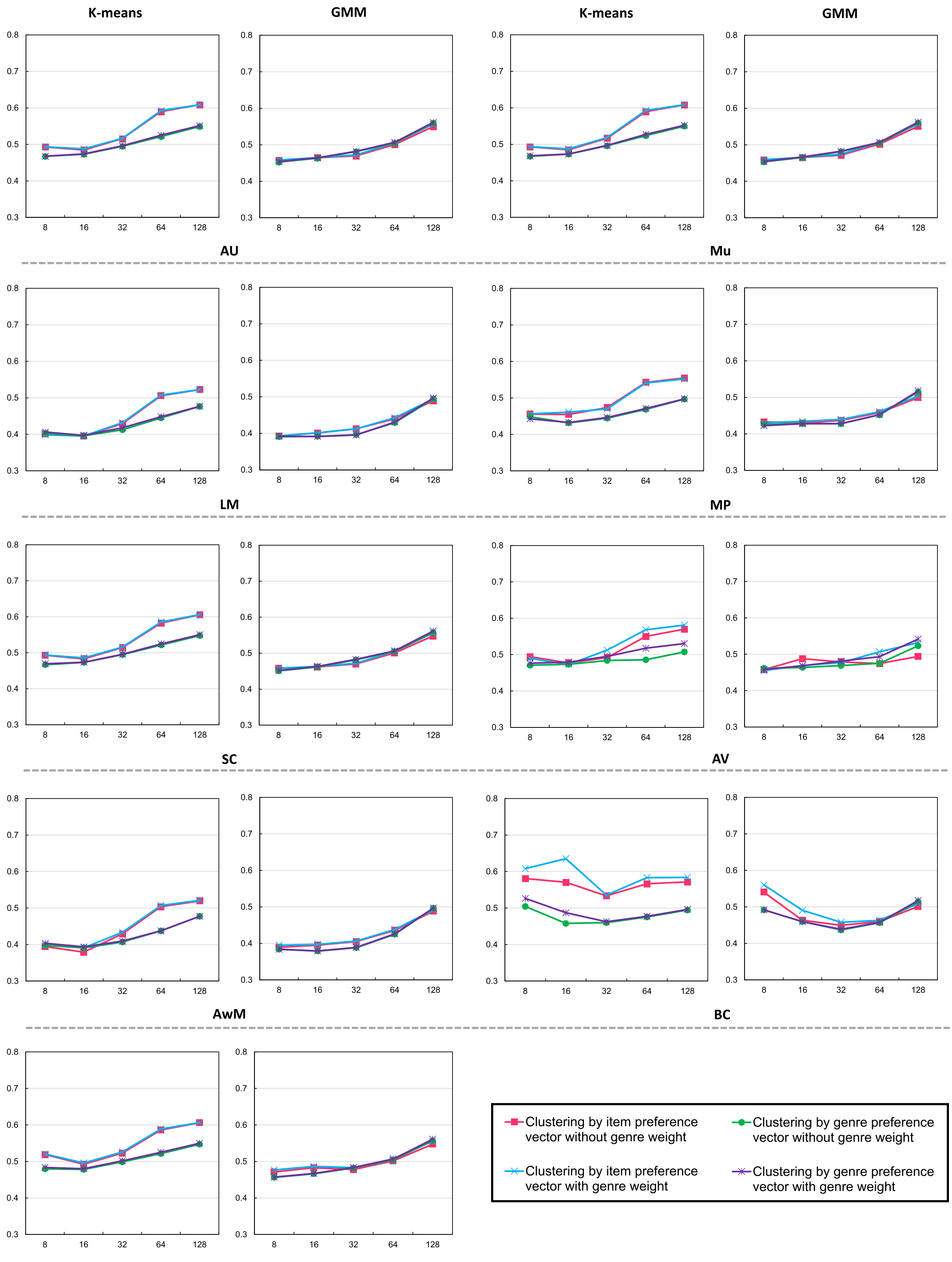


Fig. 5c. Comparative evaluation to measure effectiveness based on Last.fm (n corresponds to 20, x-axis and y-axis are NDCG and number of groups).

in MovieLens 100 k and about a hundred to a thousand times faster in Last.fm. Therefore, we verify that using the genre preference vectors can reduce the time required to form groups. Comparing the two clustering algorithms, the time complexity of k-means is linear [i.e., *O(vkd)*] (Arthur & Vassilvitskii, 2006), whereas the GMM is quadratic [i.e., *O* (vkd^2)] (Bishop, 2006). Therefore, clustering using GMM with the item preference vector as an input value took a particularly long time. In the case of MovieLens 25 M, this dataset has a large number of items, so we set the 'covariance_type' which is one of the parameters of 'sklearn. mixture.GaussianMixture' as 'diag' rather than as default 'full'. Therefore, the time efficiency of GMM in this case cannot be directly compared with the other datasets and the k-means.

We measured the effectiveness of the group recommender system based on genre preference by NDCG because it is the most popular evaluation metric for group recommendation (Baltrunas et al., 2010). In group recommender systems, NDCG is calculated based on two assumptions: 1) high-ranked items are more useful to the group than low-ranked items, and 2) the lower the number of low-ranked items, the more useful items are to the group. To measure NDCG, we calculate both the DCG and the ideal DCG (IDCG). DCG is the total gain accumulated for the top-ranked n items and is defined as per Eq. (13).

$$DCG_g = \frac{1}{|g|}\sum_{u\in g}\left(r_{u,t_1} + \sum\nolimits_{j=1}^{n}\frac{r_{u,t_j}}{\log_2 j}\right) \tag{13}$$

$t_1, \cdots, t_n$ represents the index of ranked items based on $p_{g,i}$. Therefore, r_{u,t_j} represents the actual rating of user u for the jth ranked item. In this study, we set n as 10 and 20. IDCG is an ideal case of DCG; in other words, it is the DCG of the perfect ranking. This can be defined as per Eq. (14).

$$IDCG_g = \max\left(DCG_g\right) \tag{14}$$

Finally, NDCG is calculated by dividing DCG by IDCG. This is defined as per Eq. (15).

$$NDCG_g = \frac{DCG_g}{IDCG_g} \tag{15}$$

Figs. 4a, 4b, 4c (i.e., $n = 10$) and 5 (i.e., $n = 20$) show the overall effectiveness of the group recommender system based on the genre preference or the item preference when the datasets are (a) MovieLens 100 k, (b) MovieLens 25 M, and (c) Last.fm.

We analyzed the results of the effectiveness evaluation based on four criteria:

Is there a significant difference in accuracy when clustering using genre preference vectors ($r_{u,gen}$) in group recommendation compared to using item preference vectors?

Is the accuracy of using a new item preference ($p_{u,i}$) as the input value of the group recommendation system higher than that when using an existing preference ($r_{u,i}$)?

Which aggregation method is more accurate when using genre information?

For which clustering algorithm (k-means or GMM) is the accuracy of the group recommendation system higher?

First, we compared the accuracy of the group recommender systems that use genre preference vectors or item preference vectors for group clustering. As shown in Figs. 4a, 4b, 4c and Figs. 5a, 5b, 5c, the difference between the overall NDCG values of group clustering based on two vectors is not very significant or is almost similar in MovieLens 100 k and Last.fm datasets. In particular, if the number of users in a group is low in MovieLens 100 k, then the accuracy of the group recommendation is higher when group clustering is performed based on the genre preference vector (i.e., in cases where "$n = 10$ and $k = 128$" and "$n = 20$ and $k = 128$"). In the case of Last.fm, when using k-means, the overall NDCG value is low, but the clustering cost is much more efficient. In MovieLens 25 M, when the number of groups is small (i.e., "$k = 8$ and $k = 16$), the difference between accuracies is almost similar. And, in the other cases (i.e., "$k = 32$, $k = 64$, and $k = 128$), the overall accuracy of the method clustered by the genre vector is lower than the method clustered by the item vector. However, compared to the time inefficiency of clustering, the difference in accuracy is relatively insignificant. Therefore, we conclude that the overall performance of the genre preference vector is better when considering both time complexity (i.e., efficiency) and accuracy (i.e., effectiveness).

Second, we compared the accuracies when the proposed preference ($p_{u,i}$) and the existing preference ($r_{u,i}$) are used in group recommendations. As shown in Figs. 4a, 4b, 4c and Figs. 5a, 5b, 5c, the proposed preference that considers the genre weight is accurate in MovieLens 100 k except for a few cases such as "BC in $n = 10$ and $k = 64$" and "BC in $n = 20$ and $k = 64$" for k-means, and "AwM in $n = 10$ and $k = 32$," "BC in $n = 10$ and $k = 128$," "BC in $n = 20$ and $k = 64$," and "BC in $n = 20$ and $k = 128$" for GMM. In MovieLens 25 M, the accuracy of the proposed preference is higher than the existing one in all cases. According to these results, we can analyze the advantage of subdividing the explicit ratings based on genre weight from two aspects: 1) the accuracy of recommendations increases, and 2) we can measure the evaluation metric, especially NDCG, more accurately. However, in the case of Last.fm, there is no significant difference in accuracy between both ratings. In other words, NDCG values are almost similar. Last.fm is an implicit rating system, so ratings are not skewed in a specific value. Therefore, the influence of genre weight is insignificant compared to MovieLens.

Third, we compared the accuracies of the aggregation methods based on the genre preference value. In the case of MovieLens 100 k, the UL, Mu, BC, and LM methods outperform the other methods, except when the accuracy of the AwM method was the highest (i.e., "$n = 20$ and $k = 128$ in genre-based clustering using the proposed preference value having genre weight"). For the UL and Mu methods, it was proven that their effectiveness was higher than those of other aggregation methods in existing studies (Masthoff, 2015; Seo et al., 2018). Similarly, in this study, we verified that the use of these methods in a group recommender system based on genre preference guarantees high effectiveness. BC is an aggregation method that was highly effective in existing studies (Baltrunas et al., 2010; Boratto et al., 2016; Seo et al., 2018) but does not guarantee the highest effectiveness. However, we used the weighted preference value in this study so that the preference values of users for items could be sorted more precisely. This means that the accuracy of the method that ranks the item is increased; therefore, the BC method can show high effectiveness in the group recommender system based on our preference value with genre weight. In addition, the LM method outperformed the other methods when using GMM, although its accuracy was slightly lower than that of the UL and BC methods. In a previous study (Baltrunas et al., 2010), the performance of the LM method was observed to be good when clustering groups using WUSN. On the other hand, in MovieLens 25 M, the number of items increased compared to MovieLens 100 k. So, the popular movies had a lot of influence on the results of the group recommendation. In other words, the overall performance of the methods of giving weight to items with a large number of ratings in a group, such as SC, AV, and, AU were high. Furthermore, we confirmed that UL and Mu showed high performance as in MovieLens 100 K. However, LM and BC did not because they do not consider the popularity of items. Finally, in Last.fm, the NDCG values of all methods are not significantly different, but BC has particularly high accuracy. Since Last.fm is an implicit rating system, the user ratings for each item are almost different. Therefore, when sorting according to the rating, the ranking score can be measured more accurately than the explicit rating because tie processing is not required. As shown in the results of the three datasets, the best aggregation methods are different. In other words, we proved an optimal aggregation method for group recommendations to ensure high accuracy across all domains did not exist yet.

Finally, we compared GMM and k-means. Unlike k-means, GMM was not used to cluster groups in existing group recommendations despite its efficiency. In this study, we utilized GMM as well as k-means and

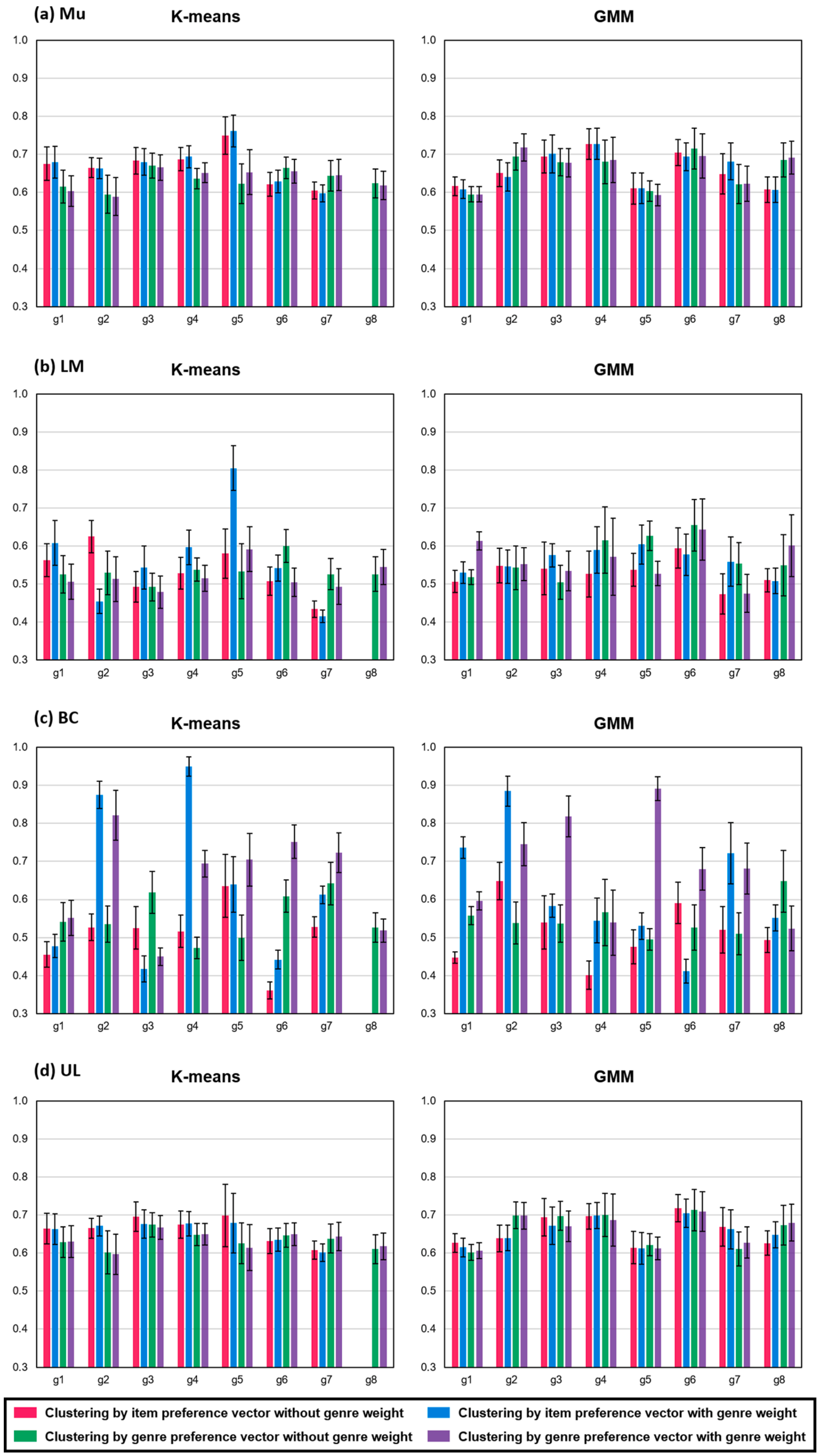


Fig. 6. The average NDCG@10 for each group with a 95% confidence interval based on a MovieLens 100 k u1 dataset (The x-axis and y-axis mean each group and NDCG).

analyzed the accuracy of the aggregation method based on GMM. As shown in Figs. 4a, 4b, 4c and Figs. 5a, 5b, 5c, we confirmed that GMM was slightly more accurate than k-means in MovieLens 100 k and 25 M, but there was no significant difference. In addition, in the case of Last.fm, the accuracy is almost similar in both cases. Therefore, clustering algorithm did not have a significant impact on the overall performance in group recommendations.

Following the efficiency (i.e., Fig. 3) and effectiveness (i.e., Figs. 4a, 4b, 4c and Figs. 5a, 5b, 5c) experiments, we analyzed the distribution of users' satisfaction within a group to verify the correlation between a group's NDCG value and the individuals' NDCG values in that group. We only considered a case of forming 8 groups using an *u1 dataset* of MovieLens 100 k and measured the average NDCG for each group with a 95% confidence interval, as shown in Fig. 6. In addition, we analyzed only the cases of Mu, LM, BC, and UL, which showed excellent performance in MovieLens 100 k.

First, we confirmed that the NDCG distribution of the GMM was more even than the k-means. In k-means clustering based on item preference vector, there is only one user in g8 (i.e., 8th group). In that case, we could not measure the mean and standard deviation because the aggregation method cannot be applied in g8. According to these results, we verify that GMM is more suitable than k-means for the group recommender system.

And then, we compared the results of Mu, LM, BC, and UL. Mu and UL had higher NDCG values and more even distribution compared to other methods. The average NDCG among groups and the users' NDCG within each group are evenly distributed. Mu gives weight to high rating and penalty to low rating due to the nature of multiply. This characteristic guarantees high performance with evenly distributed. Further, we can explain the even distribution of UL's NDCG results because UL uses a deviation as an essential factor. And then, there are wide deviations in the NDCG values of BC compared to other methods. BC is a ranking-based method, so individual user tendency influences group recommendation results. As a result, the average of each group's NDCG and the users' NDCG in a group have a high deviation. Finally, LM did not have high performance, but when clustered with GMM, the deviation appeared evenly. LM is a method of considering the standard deviation despite not using it directly. The lowest rating of an item within a group being high means the average is high with low variance.

6. Conclusions

In this study, we proposed a group recommender system based on genre preference. First, we defined the genre preference vector and formed groups using this vector. In existing studies, group clustering was performed using the item preference vectors, and therefore the cost of clustering the groups was higher than when compared to that when using the genre preference vectors. We evaluated the efficiency by measuring the time required for group clustering and verified that the time required was significantly reduced when using the genre preference vector. In addition, we concluded that the accuracy of the case where groups are clustered by genre preference was not significantly different from the accuracy of the existing clustering method based on the item preference. Second, we measured the genre weight by using the genre preference vector and utilized this weight as a weighted value of the item preference. As a result, the item preference of users can be fine-grained such that it is possible to differentiate the ratings which are skewed in positive. In addition, we validated that using our proposed preference resulted in higher accuracy than using the existing preference in group recommendation based on explicit preference.

We will propose a new aggregation method suitable for a genre-based group recommender system to increase the effectiveness and accuracy in our future studies. In existing studies, including ours, it is difficult to cluster groups on a highly large-scale matrix data set. Therefore, we will also design an efficient group clustering method for a large-scale matrix to apply it to the actual recommendation service. Finally, we will study how to reduce the distribution of users' accuracy within a group and how to normalize aggregation methods to optimize the group recommendation results with other metrics such as RMSE.

CRediT authorship contribution statement

Young-Duk Seo: Formal analysis, Investigation, Methodology, Writing - original draft. **Young-Gab Kim:** Funding acquisition, Project administration, Supervision, Writing - review & editing. **Euijong Lee:** Conceptualization, Resources, Visualization. **Hyungjin Kim:** Data curation, Validation, Software.

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.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1F1A1062480 and NRF-2021R1G1A101097111) and in part by Institute of Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2019-0-00231, Development of artificial intelligence based video security technology and systems for public infrastructure safety and 2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University)). Finally, this work was supported by INHA UNIVERSITY Research Grant.

References

Amatriain, X., Jaimes, A., Oliver, N., & Pujol, J. M. (2011). Data Mining Methods for Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. Kantor (Eds.), *Recommender Systems Handbook* (pp. 39–71). Boston, MA: Springer.

Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Baltrunas, L., Makcinskas, T., & Ricci, F. (2010, September). Group recommendations with rank aggregation and collaborative filtering. In Proceedings of the fourth ACM conference on Recommender systems (pp. 119-126). ACM.

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

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.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46*, 109–132.

Boratto, L., Carta, S., Chessa, A., Agelli, M., & Clemente, M. L. (2009, September). Group recommendation with automatic identification of users communities. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (Vol. 3, pp. 547-550). IEEE.

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.

Choi, S.-M., Ko, S.-K., & Han, Y.-S. (2012). A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications, 39*(9), 8079–8085.

Crossen, A., Budzik, J., & Hammond, K. J. (2002, January). Flytrap: intelligent group music recommendation. In Proceedings of the 7th international conference on Intelligent user interfaces (pp. 184-185). ACM.

Dara, S., Chowdary, C. R., & Kumar, C. (2019). A survey on group recommender systems. *Journal of Intelligent Information Systems*, 1–25.

De Pessemier, T., Dooms, S., & Martens, L. (2014). Comparison of group recommendation algorithms. *Multimedia tools and applications, 72*(3), 2497–2541.

Guo, J., Zhu, Y., Li, A., Wang, Q., & Han, W. (2016). A social influence approach for group user modeling in group recommendation systems. *IEEE Intelligent Systems, 31*(5), 40–48.

Guo, Z., Zeng, W., Wang, H., & Shen, Y.u. (2019). An Enhanced group recommender system by exploiting preference relation. *IEEE Access, 7*, 24852–24864.

Hafshejani, Z. Y., Kaedi, M., & Fatemi, A. (2018). Improving sparsity and new user problems in collaborative filtering by clustering the personality factors. *Electronic Commerce Research, 18*(4), 813–836.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100–108.

Hurley, N., & Zhang, M.i. (2011). Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT), 10*(4), 1–30.

Jain, A., Nagar, S., Singh, P. K., & Dhar, J. (2020). EMUCF: enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. *Expert Systems with Applications, 161*, 113724. https://doi.org/10.1016/j.eswa.2020.113724

KIM, K., & AHN, H. (2008). A recommender system using GA K-means clustering in an online shopping market. *Expert Systems with Applications, 34*(2), 1200–1209.

Li, Q.i., Liang, N.i., & Li, E. Y. (2018). Does friendship quality matter in social commerce? An experimental study of its effect on purchase intention. *Electronic Commerce Research, 18*(4), 693–717.

Mahyar, H., Ghalebi K, E., Morshedi, S. M., Khalili, S., Grosu, R., & Movaghar, A. (2017, April). Centrality-based group formation in group recommender systems. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 1187-1196). International World Wide Web Conferences Steering Committee.

Masthoff J. (2015) Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. In: Ricci F., Rokach L., Shapira B. (eds) Recommender Systems Handbook (pp 743-776). Springer, Boston, MA.

McCarthy, J. F., & Anagnost, T. D. (1998, November). MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In Proceedings of the 1998 ACM conference on Computer supported cooperative work (pp. 363-372). ACM.

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

Park, J., & Nam, K. (2019). Group recommender system for store product placement. *Data Mining and Knowledge Discovery, 33*(1), 204–229.

Pujahari, A., & Padmanabhan, V. (2015). In *December). Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques* (pp. 148–152). IEEE.

Qian, X., Feng, H.e., Zhao, G., & Mei, T. (2014). Personalized recommendation combining user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering, 26*(7), 1763–1777.

Raykov, Y. P., Boukouvalas, A., Baig, F., Little, M. A., & Yoon, B.-J. (2016). What to do when K-means clustering fails: A simple yet principled alternative algorithm. *PLoS One, 11*(9), e0162259. https://doi.org/10.1371/journal.pone.0162259

Sacharidis, D. (2019, April). Top-N group recommendations with fairness. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 1663-1670). ACM.

Seo, Y. D. (2018). *A personalized recommender system based on friendship strength using social big data* (Doctoral dissertation). Korea University).

Seo, Y.-D., Kim, Y.-G., Lee, E., & Baik, D.-K. (2017). Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications, 69*, 135–148.

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.

Shental, N., Bar-Hillel, A., Hertz, T., & Weinshall, D. (2004). Computing Gaussian mixture models with EM using equivalence constraints. In *In Advances in neural information processing systems* (pp. 465–472).

Wang, Z., Lin, G., Tan, H., Chen, Q., & Liu, X. (2020). July). CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In *In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 219–228).

Zhao, Y.-S., Liu, Y.-P., & Zeng, Q.-A. (2017). A weight-based item recommendation approach for electronic commerce systems. *Electronic Commerce Research, 17*(2), 205–226.