Expert Systems With Applications 93 (2018) 299–312

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An enhanced aggregation method considering deviations for a group recommendation

Young-Duk Seo[a], Young-Gab Kim[b,*], Euijong Lee[a], Kwang-Soo Seol[a], Doo-Kwon Baik[a,*]

[a] Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea
[b] Department of Computer and Information Security, Sejong University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 15 June 2017
Revised 11 October 2017
Accepted 12 October 2017
Available online 13 October 2017

Keyword:
Group recommendation
Aggregation method
Upward leveling
Deviation
Average
Approval voting

ABSTRACT

The goal of a group recommendation involves providing appropriate information for all members in a group. Most extant studies use aggregation methods to determine group preferences. An aggregation method is an approach that aggregates individual preferences of group members to recommend items to a group. Previous studies on aggregation methods only consider high averages, counts, and rankings to provide recommendations. However, the most important component of a group recommendation involves ensuring that majority of the members in a group are satisfied with the recommended results. Therefore, it is necessary to consider the deviation as an important element in aggregation methods. The present study involves proposing an upward leveling (UL) aggregation method that considers deviations for group recommendations. The UL recommends items with low deviations and high averages in conjunction with frequency of positive rating counts for group members. Furthermore, the effectiveness of the UL is validated to perform a comparative evaluation with existing aggregation methods by using the normalized discounted cumulative gain (NDCG) and diversity. The results indicate that the UL outperforms all the baselines and that the deviation plays an important role in the aggregation method.

1. Introduction

Recommender systems are widely examined in various academic research area as well as for commercial purposes (Bobadilla, Ortega, Hernando, & Gutierrez, 2013). Most recommender systems that are used in e-commerce, social commerce, and social network services (SNSs) focus on providing appropriate items or interests for personal users and are termed as personalized recommendation (Seo, Kim, Lee, & Baik, 2017).

However, the advent of the era of big data makes it more complex for service providers to identify the propensity of all users in a system. In other words, the application of personalized recommendations to an actual system entails high costs of maintaining the system and preventing overheads (Boratto, Carta, & Fenu, 2016). Furthermore, most individuals prefer performing certain actions or operations in tandem with other individuals as opposed to by themselves because human beings are social animals by nature (Masthoff, 2015). For example, individuals visit a famous restaurant for lunch with their colleagues, watch a funny TV program with their family, or go to a movie with their friends. Recently, studies indicated increases in the number of individuals that form groups in online communities to share common interests including movies (O'connor, Cosley, Konstan, & Riedl, 2001; Quijano-Sánchez, Díaz-Agudo, & Recio-García, 2014), music (Chao, Balthrop, & Forrest, 2005; McCarthy & Anagnost, 1998), and travel (Ardissono, Goy, Petrone, Segnan, & Torasso, 2002; Márquez & Ziegler, 2016). Therefore, it is necessary to examine recommender systems that target a group as opposed to individuals to resolve the cost problem of personalized recommendations and reflect online group activities.

The goal of group recommendations involves a method of providing appropriate information for group members to analyze the characteristics and propensity of a group (Jameson & Smyth, 2007). Most extant studies aggregate the preferences of all group members by using methods termed as aggregation methods to determine group preferences in group recommendations (Pessemier, Dooms, & Martens, 2013). An aggregation method is the most important component for group recommendations. Therefore, the present study focuses on an aggregation method to improve group recommendation quality.

There are several aggregation methods for group recommendations (Boratto et al., 2016; Masthoff, 2015). Most existing aggrega-

* Corresponding author.
E-mail addresses: seoyoungd@korea.ac.kr (Y.-D. Seo), alwaysgabi@sejong.ac.kr (Y.-G. Kim), kongjjagae@korea.ac.kr (E. Lee), seolks@korea.ac.kr (K.-S. Seol), baikdk@korea.ac.kr (D.-K. Baik).

https://doi.org/10.1016/j.eswa.2017.10.027
0957-4174/

tion methods mainly provide recommended items for group members based on high average (Ardissono et al., 2002; Berkovsky & Freyne, 2010; Feng & Cao 2017; Liu et al., 2016; Mahyar et al., 2017; McCarthy et al., 2006; Quijano-Sánchez et al., 2014; Yu, Zhou, Hao, & Gu, 2006) or frequency of rating counts (Crossen, Budzik, & Hammond, 2002; Lieberman, Van Dyke, & Vivacqua, 1999). Although this appears reasonable, these methods do not always guarantee high quality recommendations for groups because they are unable to reflect the propensities of all users in a group. For example, if a group's deviation of preferences is high when the recommended items involve a high average and frequency of rating counts, then this increases the number of users that do not prefer the recommended results. Therefore, these methods are inappropriate for group recommendations in the fore-mentioned case. Other studies propose aggregation methods based on ranking results for users in groups such as scoring ranking results (Márquez & Ziegler, 2015; 2016) and considering the highest rank item by priority (Khoshkangini, Pini, & Rossi, 2016). However, these methods are highly dependent on high average and frequency of rating counts. Most importantly, they do not consider the deviation of all users' rates in a group. A few aggregation methods consider the deviation. However, they simply exclude extreme cases (Agarwal, Chakraborty, & Chowdary, 2017; Chao et al., 2005; Christensen & Schiaffino, 2011; Feng & Cao, 2017; McCarthy & Anagnost, 1998; O'connor et al., 2001), and thus they do not consider entire preferences of group members and result in a lower performance in groups that consist of a large number of members.

In contrast to personalized recommendations, the maximum number of users that prefer the recommended items is more important in group recommendations as opposed to providing perfectly suitable results for individual users. Thus, it is important to determine items in which the ratings of group members are evenly distributed. Therefore, it is necessary to consider the deviation of ratings in a group as the most important element in group recommendations. In this study, in order to overcome the limitations of existing aggregation methods, a new aggregation method termed as the upward leveling method is proposed for group recommendations, and it is based on the deviation as well as average and rating count. The proposed method is used to provide all group members with suitable recommendations that reflect individual tendencies.

The main contributions and goals of the study are as follows:

(1) The proposed aggregation method is termed as the upward leveling method (UL), and it provides appropriate recommendation results for group members by considering the deviation of ratings. Specifically, the deviation is suitably combined with average and approval voting methods in the UL.
(2) Groups are formed to use a k-means clustering algorithm, and each group includes several members (e.g., at least five members). Experiments performed in the fore-mentioned conditions verified that a high level of performance of the UL is observed for groups with several members.
(3) In the study, the proposed method is compared with existing aggregation methods based on normalized discounted cumulative gain (NDCG) and diversity to verify the effectiveness of the UL. The results indicate that the performance of the UL significantly exceeds that of baselines. Additionally, the deviation plays an important role in group recommendations and especially in the aggregation method.

The rest of this study is structured as follows. Section 2 describes the existing aggregation methods that are used in group recommendations and their limitations. Section 3 examines related studies based on an existing aggregation method. Section 4 explains the proposed group recommendation methodology based on the UL. Section 5 describes an evaluation framework to evaluate the performance of the UL and shows a comparative evaluation in which data from the MovieLens dataset is used. In Section 6, we discuss the limitations our study and why we are focusing on a homogeneous group rather than a heterogeneous group. Section 7 presents conclusions and future study directions.

Table 1
The users-to-items matrix used to explain existing aggregation methods.

Users	Items				
	i_1	i_2	i_3	i_4	i_5
u_1	2	3	5	4	■
u_2	3	2	5	4	3
u_3	2	■	1	3	4

Table 2
The result of the PV method according to the uses-to-items matrix in Table 1.

Users	Ranks				
	1	2	3	4	5
u_1	i_3	i_4	i_2	i_2	i_2
u_2	i_3	i_4	i_1, i_5	i_1	i_2
u_3	i_5	i_5	i_5	i_1	i_2
PV	i_3	i_4	i_5	i_1	i_2

2. Problem statements

In group recommendations, most extant studies provide recommended results for group members to aggregate the ratings of all members. These methods are termed as aggregation methods and are mainly used in group recommendations due to their effectiveness and accuracy (Pessemier et al., 2013). Previous studies used different terms for aggregation methods, such as aggregation strategy (Masthoff, 2015) and group modeling (Boratto et al., 2016).

There are several aggregation methods including additive utilitarian (Agarwal et al., 2017; Boratto et al., 2016; Kaššák et al., 2016; McCarthy, 2002), average (Ardissono et al., 2002; Berkovsky & Freyne, 2010; Feng & Cao 2017; Liu et al., 2016; Mahyar et al., 2017; McCarthy et al., 2006; Quijano-Sánchez et al., 2014; Yu et al., 2006), multiplicative (Christensen & Schiaffino, 2011), most pleasure (Boratto et al., 2016), plurality voting (Khoshkangini et al., 2016), simple count (Crossen et al., 2002), approval voting (Boratto et al., 2016; Lieberman et al., 1999), borda count (Márquez & Ziegler, 2015; 2016), copeland rule (Masthoff, 2015), most respected person (Masthoff, 2015), average without misery (Chao et al., 2005; McCarthy & Anagnost, 1998), least misery (Agarwal et al., 2017; Christensen & Schiaffino, 2011; Feng & Cao, 2017; O'connor et al., 2001), and fairness (Christensen & Schiaffino, 2011; Villavicencio et al., 2016) methods. With respect to the fairness method, two methods with the same name exist. They are classified into the following seven sections based on their properties: "Simple Computation", "High Rating Priority", "Counting the Ratings", "Ranking Priority", "Comparing the Ratings", "Based on the degree of Influence", and "Considering Deviation". Table 1 illustrates the users-to-items matrix that is used in the following sections to describe a few aggregation methods as shown in Tables 2–6. Only aggregation methods that require detail explanations (i.e., plurality voting, borda count, fairness, copeland rule, average without misery, and least misery) are shown in the tables in the next subsections and the remaining methods (i.e., additive utilitarian, multiplicative, average, most pleasure, simple count, approval voting, and most respected person) are listed in the appendix.

Table 3
The result of the BC method according to the uses-to-items matrix in Table 1.

Users	Items				
	i_1	i_2	i_3	i_4	i_5
u_1	0	1	3	2	■
u_2	1.5	0	4	3	1.5
u_3	1	■	0	2	3
BC	2.5	1	7	7	4.5

Table 4
The result of the Fa method according to the uses-to-items matrix in Table 1.

Users	Ranks				
	1	2	3	4	5
u_1	i_3	■	■	i_2	■
u_2	■	i_4	■	■	i_1
u_3	■		i_5	■	■
Fa	i_3	i_4	i_5	i_2	i_1

Table 5
The result of the CR method according to the uses-to-items matrix in Table 1.

Items	Items				
	i_1	i_2	i_3	i_4	i_5
i_1	■	−1	+1	+1	0
i_2	+1	■	+1	+1	+1
i_3	−1	−1	■	-1	−1
i_4	−1	−1	+1	■	−1
i_5	0	−1	+1	+1	■
CR	−1	−4	+4	+2	−1

Table 6
The result of the "Considering Deviation" method according to the uses-to-items matrix in Table 1.

	Items				
	i_1	i_2	i_3	i_4	i_5
AwM	■	■	■	3.67	3.5
LM	2	2	1	3	3
Fa2	2.16	2.29	2.97	3.49	3.29

2.1. Simple computation

The easiest method to aggregate the preferences of group members involves using fundamental arithmetic operations such as addition and multiplication. They are termed as "Simple Computation" methods and include "Additive Utilitarian (AU)" (McCarthy & Anagnost, 1998), "Multiplicative (Mu)" (Christensen & Schiaffino, 2011), and "Average (Avg)" (Ardissono et al., 2002; Berkovsky & Freyne, 2010; Feng & Cao 2017; Liu et al., 2016; Mahyar et al., 2017; McCarthy et al., 2006; Quijano-Sánchez et al., 2014; Yu et al., 2006). Specifically, AU and Mu correspond to methods that add and multiply all group members' preference ratings, respectively. The Avg method calculates the average of the preference ratings of the group members.

Although simple computation methods constitute easy ways to measure group preferences, certain difficulties are involved in using them as the results of group recommendation. Specifically, AU and Mu may consider low average items rated by several members as better results for groups as opposed to high average items rated by a few members. Additionally, with respect to Mu, if a few items are rated by several group members, then their group preference ratings converge to infinity and it is not possible to measure their group preference ratings. The Avg method appears to constitute the most reasonable aggregation method for group recommendation. However, it also entails a problem wherein a high average could lead to a high deviation of the recommended items' ratings, and this could subsequently increase the number of users that do not prefer the results. Furthermore, the Avg method sets the high average items rated by a few group members as the recommendation results for the group. In these cases, the Avg method is inappropriate for group recommendations.

2.2. Highest rating priority

The highest rating is an extremely important element for aggregation methods. Therefore, there are methods associated with the highest rating, such as "Most Pleasure (MP)" (Boratto et al., 2016) and "Plurality Voting (PV)" (Khoshkangini et al., 2016). They are termed as "Highest Rating Priority" methods. Specifically, the MP method considers the highest rating in group as a group preference rating for the item. The MP method is not preferred for systems in which the scope of the rating (i.e., 0–5) is fixed. When there are several members in group, the group preference ratings are identical to the highest rating of that system for almost all items. Therefore, it is difficult for the MP to grasp the overall preference of the group.

The PV method is another way to consider the highest rating as important. In the case of the PV method, the highest rated items for each user are initially selected. Subsequently, the item that exhibits the highest rating for a majority of the members in the group is selected as the most preferred item. This method is repeated until the list is completed with the exception of the items that were already selected in the list of preferred items. An example of PV is shown in Table 2. The PV method alleviates the problem of MP. However, the PV method does not consider negative preferences, and it leads to the infeasibility of identifying an even distribution of ratings. Furthermore, the PV calculation entails considerable time when compared with other methods because of the sorting process involved.

2.3. Counting the ratings

Most users rate items in which they are interested. Therefore, counting preference ratings is an important factor to determine group preference ratings. There are two methods that consider the counts, namely "Simple Count (SC)" (Crossen et al., 2002) and "Approval Voting (AV)" (Boratto et al., 2016; Lieberman et al., 1999) methods. They are termed as "Counting the Ratings" methods. Specifically, the SC method simply counts all items rated by users while the AV method only counts positive ratings that exceed a specified threshold.

In the case of the SC method, a problem occurs when a few items involve frequency of negative ratings. The AV method resolves this problem and it appears reasonable because positive ratings outnumber negatives with respect to most systems in practice (Bobadilla, Serradilla, & Bernal, 2010). However, it is difficult to analyze the overall preferences of a group if all negative ratings are completely excluded.

2.4. Ranking priority

Generally, the most preferred item corresponds to the item with the highest ranking in the rated items list for users. Therefore, extant studies propose aggregation methods that place a greater priority on ranking. Examples of these methods include "Borda Count (BC)" (Márquez & Ziegler, 2015; 2016) and "Fairness (Fa)" (Masthoff, 2015) methods. They are termed as "Raking Priority" methods. The BC method scores ratings based on the ranking results (i.e., the highest ranking scores $n-1$ and the lowest scores 0 in the n items), while the Fa method determines the item with the highest ratings provided by users by means of a rotation. Examples of BC and Fa are shown in Tables 3 and 4, respectively.

The BC and FA methods focus on the ranking of the ratings. However, they are highly dependent on high average and counts. Most importantly, they do not consider the deviation.

2.5. Comparing the ratings

A relatively preferred item is considered as more appropriate than other items. The "Copeland Rule (CR)" method (Masthoff, 2015) measures group preference ratings by calculating the relative importance of items. As shown in Table 5, the i_1 score for i_2 corresponds to $+1$ and the other score corresponds to -1 because more members prefer i_1 when compared to i_2. However, the time taken by CR to calculate group preference ratings is the highest because it compares all ratings between all users.

2.6. Based on the degree of influence

An influential individual affects the entire network on systems (Lin, Xie, Guan, Li, & Li, 2014). A few methods follow the decision of an influential individual. Examples of these methods include the "Most Respected Person (MR)" method (Masthoff, 2015). In the MR method, it is extremely important to determine an influential person in a group. However, the determination of group preferences by only using a single member is not considered as a suitable method in group recommendations.

2.7. Considering deviation

The most important consideration in group recommendations relates to the maximum number of members that prefer recommended items as opposed to providing perfectly suitable results for each member. Therefore, the distribution and deviation of the group members' ratings are considered as important. Specifically, these methods include "Average without Misery (AwM)" (Chao et al., 2005; McCarthy & Anagnost, 1998), "Least Misery (LM)" (Agarwal et al., 2017; Christensen & Schiaffino, 2011; Feng & Cao, 2017; O'connor et al., 2001), and "Fairness (Fa2)" (Christensen & Schiaffino, 2011; Villavicencio et al., 2016) methods and are termed as "Considering Deviation" methods. Table 6 shows an example of "Considering Deviation" methods.

The AwM method is similar to the Avg method although the method does not calculate an average for an item as a group preference rating if at least one member's rating is lower than the threshold rating. The LM method selects the minimum rating in a group as a group preference rating. The high minimum rating implies that the average of ratings is high with a small deviation in the group. Therefore, the LM method is included in the "Considering Deviation" method. The Fa2 method calculates the preference of a group by considering the deviation as a weighted value of the average. The "Considering Deviation" methods constitute the most frequently used methods in group recommendation due to their high performance. However, they simply exclude extreme cases and do not consider the overall deviation of group members' ratings. Therefore, they lead to poor quality when a group includes several members.

3. Related work

Extant studies suggest different scenarios for group recommendations in various areas such as movies (Agarwal et al., 2017; Boratto et al., 2016; Christensen & Schiaffino, 2011; Feng & Cao, 2017; Kaššák, Kompan, & Bieliková, 2016; Liu et al., 2016; Mahyar et al., 2017; O'connor et al., 2001; Quijano-Sánchez et al., 2014; Villavicencio et al., 2016), music (Chao et al., 2005; Christensen & Schiaffino, 2011; Crossen et al., 2002; McCarthy & Anagnost, 1998), tours (Ardissono et al., 2002; Márquez & Ziegler, 2015; 2016; McCarthy et al., 2006), restaurants (Khoshkangini et al., 2016; McCarthy, 2002), recipes (Berkovsky & Freyne, 2010), TV Program (Yu et al., 2006), and browsing (Lieberman et al., 1999). Although various group recommender systems exist in several domains, optimal aggregation methods differ for each proposed scenario. In other words, a single optimal aggregation method that exhibits a high performance in all scenarios does not exist.

Table 7 represents the classification of existing group recommendation systems. They are classified into the following four categories: system name, aggregation method used in each system, group size used in the experiment, and experimental domain.

MusicFX (McCarthy & Anagnost, 1998) plays appropriate music for members who currently use a fitness center by considering their preference information for a music genre. The system classifies the preference of each member for a music genre into five levels (i.e., $+2$, $+1$, 0, -1, -2), and measures the squared sum of preferences and the probability that a music genre is selected to identify the preference of members in the fitness center. Furthermore, MusicFX uses the AwM method, in which a music genre is selected only when the preferences of all group members exceed a threshold level. Lieberman et al. (1999) proposed the Let's Browse system that is helpful for web browsing for a group. They compared a web page with a user profile and calculated the extent to which a page matches a profile. This recommendation method is similar to the AV method because it considers the number of matching scores for group members that exceed a certain threshold. O'connor et al. (2001) proposed the PolyLens recommender system that extends the MovieLens scenario to fit a group recommender system based on the collaborative filtering (CF) method. They used the LM as an aggregation method and conducted a survey to verify the satisfaction of users for the system. The Flytrap (Crossen et al., 2002) recommender system consists of three main elements to provide recommended songs for group members, namely track information recorder, genre network, and voting mechanism. First, the system identifies the tracks that users frequently listen to and stores the tracks' information in a track information recorder. Subsequently, it measures similarity between music genres to construct a semantic network by extracting metadata of tracks to use ID3 tags in MP3 files. Finally, a voting mechanism that is similar to the SC method allocates a weight for a track with a high rating to measure the group preferences for a track.

The INTRIGUE (Ardissono et al., 2002) corresponds to a web-based adaptive system that provides various type of information (such information related to accommodation and food) related to tourist attractions for a tourist group. In the system, a group model that represents the characteristics of a group is represented by the following three elements: characteristics, preferences, and group information. Characteristics corresponds to the characteristics of group members and includes age, background, mobility, and interest. Preference represents the average preference of a group (i.e., the Avg method) for tourist attractions, and group information represents the number of members and the relationship among members in a group. The Pocket RestaurantFinder (McCarthy, 2002) is an application that recommends restaurants to individuals desirous of sharing a meal. The group preference for a restaurant is calculated by aggregating the preferences of each member based on the AU method and involved measurements that consider the location, price, cuisine, and amenities. The adaptive radio (Chao et al., 2005) is a music server that broadcasts suitable songs for groups. They used a negative preference method by assuming that it is easier to determine the songs disliked by users as opposed to determin-

Table 7
Classification of existing group recommender systems.

System	Aggregation method	Group size*	Domain
MusicFX (McCarthy & Anagnost, 1998)	AwM	Medium	Music
Let's Browse (Lieberman et al., 1999)	AV	n/a	Browsing
PolyLens (O'connor et al., 2001)	LM	Small	Movie
Flytrap (Crossen et al., 2002)	SC	Small	Music
INTRIGUE (Ardissono et al., 2002)	Avg	Small	Tour
Pocket RestaurantFinder (McCarthy, 2002)	AU	Small	Restaurant
Adaptive Radio (Chao et al., 2005)	Negative Preference (Without Misery)	Small	Music
CATS (McCarthy et al., 2006)	Avg	Small	Tour
Yu et al. (2006)	Avg	Small	TV Program
Berkovsky and Freyne (2010)	(Weighted) Avg	Small	Recipe
jMovieGroupRecommender & jMusicGroupRecommender (Christensen & Schiaffino, 2010)	MU, LM, Avg, Fa2	Medium	Music, Movie
HappyMovie (Quijano-Sánchez et al., 2014;2015)	Avg	Small	Movie
Hootle + (Márquez & Ziegler, 2015; 2016)	BC	Small	Tour
Boratto et al. (2016)	AU, AV, BC, LM, MP	Large	Movie
CoGrec (Liu et al., 2016)	Avg	Large	Movie
Kaššák et al. (2016)	AU	Small	Movie
PUMAS-GR (Villavicencio, 2016)	Fa2	Small	Movie
SaCARS (Khoshkangini et al., 2016)	PV	Small	Restaurant
Agarwal et al. (2017)	AU, LM	Large	Movie
Feng and Cao (2017)	(Weighted) Avg, LM	Small	Movie
Mahyar et al. (2017)	(Weighted) Avg	Medium	Movie

* indicates the group size in the experiments that classified small, medium, and large (Small := # members in group ≤ 10, Medium := 10 < # members in group ≤ 100, Large := 100 < members in group).

ing the song liked by users. The negative preference method never considers positive preferences and instead filters negative information in a group. The aggregation method in the Adaptive Radio corresponds to a without misery aspect. The CATS (McCarthy et al., 2006) corresponds to a collaborative group recommender system that provides information related to a ski-trip. A personal individual model in CATS stores the preferences of each member in a group, and this is aggregated to measure group preferences. The group preference value is calculated by using the Avg method and is managed in a group user model. This system immediately reflects the feedback of group members and updates a personal individual model as well as a group user model to provide recommendations for each member and the group, respectively. Yu et al. (2006) proposed a TV program recommender system by merging all group members' profiles to recommend TV programs that satisfy all members. Each member's profile in their system is represented by a preference vector that consists of features measured as 1 (positive), 0 (neutral), and −1 (negative). The Avg of the normalized vector for each member is used as the group profile vectors.

Berkovsky and Freyne (2010) suggested recipe recommendations for families that constitute special groups in the group recommendation area. They measured the relative influence of family members and then calculated the preference of a family for a recipe through a weighted average value. Therefore, they used the weighted Avg method. Christensen and Schiaffino (2011) presented entertainment group recommender systems, such as jMusicGroupRecommender and jMovieGroupRecommender based on a GroupRecommendation framework. They used several group recommendation methods including aggregation of the result of personalized recommendation, aggregation methods, and construction of a group preference model. Specifically, in the fore-mentioned systems, Mu, Avg, LM, and Fa2 methods are selected for aggregation methods, and the group preference model represents the preference of a group by analyzing the profiles of group members. Quijano-Sánchez et al. (2014) implemented a social recommendation model with various social factors on SNS as a HappyMovie application. They aggregated the preferences of members by using the Avg method, reliability among members, and influence among members in a group to measure the group preference. They also expanded the function of the HappyMovie application to add un-profiled user modeling and hierarchical relationship group modeling, and performed experiments with actual family members (Quijano-Sánchez, Recio-García, & Díaz-Agudo, 2015). Hootle+ (Márquez & Ziegler, 2015; 2016) investigated a system that recommends suitable hotels for groups, and measures a group preference based on the BC method. In the system, it is possible to change group recommendation results by supporting discussion and negotiation processes among members in the groups. If a member suggests a hotel, then the remaining members votes to determine the acceptance of the suggested hotel. Boratto et al. (2016) simply measured the performance of the aggregation methods, such as the AU, AV, BC, LM, and MP methods, for a few group members as well as several group members, although they did not propose a new group recommender system.

Liu, Wang, Wu, Zeng, Shi, and Zhang (2016) proposed the CoGrec system that extracts the profile of users based on non-negative matrix factorization. They detected overlapping communities and performed an aggregation method from the viewpoint of an overlapping group as opposed to a single group. Their aggregation method considered the average of the group preference (i.e., the Avg method), influence of members in overlapping groups, and the advantage of overlapping groups for members. Kaššák et al. (2016) proposed a hybrid group recommendation method that combined content-based (CB) and collaborative-based (CF) methods. They measured group preferences by aggregating the CB and CF methods based on the AU method and combined them to calculate hybrid scores. The PUMAS-GR (Villavicencio et al., 2016) is a group recommender system that is based on a multi-agent system. Each agent behaves on behalf of a user and includes information such as the preferences of users and ranking results. It forms groups by using negotiation among agents based on monotonic concession protocol (MCP). The preference of a group is measured by using the average preference with the standard deviation to provide suitable recommendation results, and this is similar to the Fa2 method. The SaCARS (Khoshkangini et al., 2016) considers contextual information that affects the preference of a user for a group recommendation. It uses CP-net formalism and Hyperspace Analogue to Context (HAC) to model user preferences and various contextual information, respectively. Furthermore, it involves different weights based on the importance of members in a group and aggregates the features of interest in a group by using an aggregation method that is similar to PV. Agarwal et al. (2017) proposed a hungarian aggregated method and least misery with priority method by extending the aggregated voting method, and the proposed method is similar to the AU method and LM methods, respectively. Their results indicate that the performances of the proposed methods exceeded those of existing methods. Feng and Cao (2017) detected the inherent relationship among group members and items by using a random walk with restart (RWR). The aim of the study involved improving performance and solving the data sparsity problem. The RWR was used to detect the relationship between data, and the group preference was measured by using weighted Avg and LM methods. Mahyar, Ghalebi K, Morshedi, Khalili, Grosu, and Movaghar (2017) measured the influence of members in a group by using centrality in graph theory (especially betweenness centrality) and calculated group preferences through a weighted Avg method. Most extant studies did not consider the deviation as an important element. However, a few studies used the deviation (Christensen & Schiaffino, 2011; Villavicencio et al., 2016). Consequently, most studies conducted experiments in the small group size environments because they did not consider the deviation.

MusicFX (McCarthy & Anagnost, 1998), jMovieGroupRecommender & jMusicGroupRecommender (Christensen & Schiaffino, 2011), Boratto et al. (2016), CoGrec (Liu et al., 2016), Agarwal et al. (2017) and Mahyar et al. (2017) proposed large group size-based group recommender systems. However, a few studies (Agarwal et al., 2017; Christensen & Schiaffino, 2011; McCarthy & Anagnost, 1998) did not use objective evaluation metrics that are generally used in group recommendation, such as precision, recall, NDCG, MAE, and RMSE, and simply used surveys or their own metrics. Christensen and Schiaffino (2011) and Boratto et al. (2016) conducted experiments to compare the performance of various aggregation methods in environments based on an extremely high group size although they did not propose new aggregation methods. Furthermore, the performance of a system was not good (Liu et al., 2016) or there are no comparative experiments with various aggregation methods in a large group size (Agarwal et al., 2017; Mahyar et al., 2017). In contrast to existing methods, the UL method considers the deviation as the most important element. In the present study, comparative experiments ranging from a small scale to a large group size scale were performed, and the results exhibited that the performance of the proposed method exceeded those of most extant aggregation methods.

4. Upward leveling method for group recommendation

From the group recommendation standpoint, the aim of the present study involves providing suitable recommendation results to satisfy the maximum possible number of members in a group. In order to achieve this aim, the UL method is proposed to calculate the predicted ratings of groups. The deviation is considered as the most important element in the UL method, and it is properly combined with the Avg and AV methods. Additionally, raw data (i.e., MovieLens) is converted into an users-to-ratings matrix, and the groups are detected by using a k-means clustering algorithm prior to calculating the predicted ratings of the groups.

4.1. Framework for group recommendation based on the upward leveling method

As shown in Fig. 1, a framework for group recommendation is proposed to determine the relevant items for the groups.

The framework consists of three steps, namely “Converting raw data into an users-to-ratings matrix”, “Identification of groups,” and “Calculate the predicted ratings of group”. First, raw data, such as MovieLens, are converted into an users-to-ratings matrix. Additionally, the range of ratings is normalized from 0 to 1 for all data such that it can be applied to the UL based group recommender systems. The normalization of the rating is required to calculate the proposed aggregation method and align the range of rating with the same range. In the study, MovieLens 100k[1] is used and corresponds to the most representative explicit data set in recommender systems. The MovieLens data set includes a general range of ratings ranging from 1 to 5 in recommender systems and a movie in which a rating closer to 5 represents an extremely preferred movie. All ratings are converted into ratings ranging from 0.2 to 1 by dividing all numbers by 5 to measure the predicted ratings based on the UL.

In the second step, the groups for all users are identified such that the k-means clustering algorithm (Hartigan & Wong, 1979) is used. Each group consists of users with similar interests. In a group recommendation, identification of groups is indispensable to determine interests or preferences of groups (Boratto, & Carta 2010). Although a few studies simply use an established group (Quijano-Sánchez et al., 2014; 2015), most studies automatically detect groups based on the preference information of users in a community. There are at least two methods to automatically determine the groups as follows: 1) detecting the groups after calculating similarities between all users such that the Pearson correlation coefficient (Baltrunas, Makcinskas, & Ricci, 2010) can be used, and 2) simply using a clustering algorithm such as *k*-means (Boratto et al., 2016), and hierarchical clustering methods (Cantador, & Castells, 2011). The former method may correspond to a more precise method to determine the groups. However, the latter method is more suitable for the purpose of group recommendations. With respect to time complexity, the former method is not significantly different when compared with the personalized recommendation method, and it leads to the disappearance of the advantage of group recommendation. Therefore, the latter method is selected to determine the groups in a faster manner. Additionally, the k-means clustering algorithm is used because it corresponds to the most popular clustering algorithm and can be applied to almost all types of data (Boratto et al., 2016). Finally, the predicted ratings of each group are calculated to use the UL. Section 4.2 describes the proposed aggregation method in detail.

4.2. Upward leveling method

Generally, “the maximum average within the group” and “the number of positive ratings that exist in the group” are considered as important elements for existing aggregation methods because they appear reasonable with respect to aggregation methods. However, they are not always suitable for all group members as previously discussed in Section 2. The element corresponding to “how even the ratings are distributed within the group?” is considered as the most important element in group recommendation to determine the items that satisfy the maximum possible number of group members. In conclusion, the UL method is proposed to combine the deviation with a mainly used aggregation method (i.e., Avg and AV methods). Additionally, it is necessary for all elements to process the same range of values (i.e., from 0 to 1).

First, the Avg method is used as an element of the UL. The Avg method is representatively used among the existing aggregation methods (Masthoff, 2015; Pessemier et al., 2013), and it calculates the average of ratings for items with respect to all group members. The range of Avg is from 0 to 1 such that the Avg method is used without modifying the formula.

[1] http://grouplens.org/.

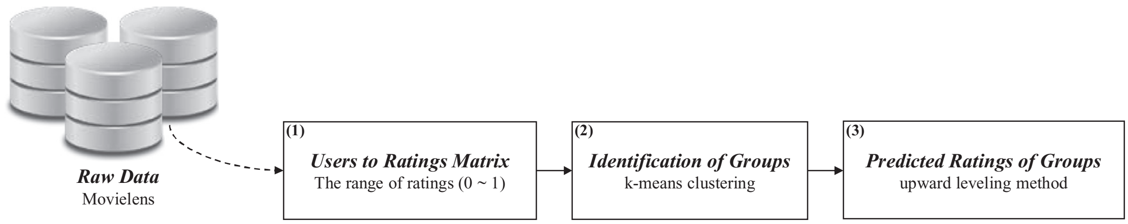

Fig. 1. A framework for the proposed group recommendation.

The AV is also considered as an element of the UL (Boratto et al., 2016; Lieberman et al., 1999) because an increase in the number of positive ratings for an item implies an increase in the number of group members that prefer the specific item. Furthermore, the method compensates for the shortcomings of the Avg that occur when the number of ratings in the group is low. However, the range of AV is not specified, and its maximum value corresponds to the total number of members in the group. Therefore, the range is modified such that it can be applied to the UL, and the AV is normalized by a min-max normalization. In the MovieLens data set, the threshold is set as 4 because 4 and 5 are considered as positive ratings (Bobadilla et al., 2010).

As previously discussed, the high Avg and AV items are not always suitable for all group members. Therefore, it is necessary to recommend items for group members with a low deviation and high Avg and AV values. This combination makes it possible to provide satisfactory recommendation to the maximum possible number of group members. In the UL method, the mean square deviation (MSD) is utilized to calculate the deviation of preference ratings for items in groups. If $\overline{r_i}$ denotes the average ratings for item i by all members in a group, and N denotes the number of members who vote i, then $MSD_{G,i}$ that corresponds to the deviation of item i by group G is defined by Eq. (1) as follows:

$$MSD_{G,i} = \frac{\sum_{u \in G} (\overline{r_i} - r_{u,i})^2}{N} \tag{1}$$

A low value of $MSD_{G,i}$ indicates that the distribution of ratings for item i is even in group G. Therefore, a decrease in the value of $MSD_{G,i}$ leads to an improvement in the results. This is in contrast to the AV and Avg methods. Therefore, it is necessary to modify $MSD_{G,i}$ for an element of UL. The deviation element $Dev_{G,i}$ is defined in Eq. (2) as follows:

$$Dev_{G,i} = 1 - MSD_{G,i} \tag{2}$$

The ultimate objective of UL involves calculating the predicted preference ratings of group for items ($p_{G,i}$). The UL method consists of a combination of $Avg_{G,i}$, $AV_{G,i}$, and $Dev_{G,i}$. Additionally, $p_{G,i}$ is calculated as a weighted sum of the fore-mentioned values and is defined in Eq. (3) as follows:

$$p_{G,i} = \alpha Avg_{G,i} + \beta AV_{G,i} + \gamma Dev_{G,i} \quad (\alpha + \beta + \gamma = 1) \tag{3}$$

5. Experiments and evaluation

In this study, a comparative evaluation is performed to verify the superiority of the UL relative to the baseline aggregation methods. Thus, NDCG and diversity are used for the evaluation metric, and a cross validation is performed to validate statistically significant results. Additionally, the importance of Dev in the UL is verified to compare the results based on the change in the Dev weight (γ).

5.1. Data set

The MovieLens 100k data set shown in Table 8 is used in the study.

Table 8
Data sets in the experiments.

	Domain	#users	#items	Type of ratings
MovieLens100k	Movie	943	1682	Explicit (1–5 ratings)

Specifically, MovieLens corresponds to a recommender system for movies that is considered as a representative recommender system. The MovieLens data set includes typical types of ratings in the recommender system corresponding to explicit ratings for a movie by users. The superiority of the UL method in terms of the explicit information is verified to use MovieLens data set.

5.2. Evaluation setup

In the study, a random subsampling cross validation is conducted to evaluate the UL method. The process of comparative evaluation for the aggregation methods is shown in Fig. 2. The process automatically detects the group and calculates predicted ratings of groups. Furthermore, it measures the NDCG value of all aggregation methods including the UL method.

First, the set of items is randomly divided into a test set and a training set in the random subsampling cross validation step. Additionally, 20% of the items is considered as a test set and 80% of the items is considered as a training set. The division of the items into the test and training sets is followed by detecting the groups to use a k-means clustering algorithm based on the set of training items. This is followed by calculating the predicted preference ratings of groups based on all aggregation methods including the UL by using the set of test items. In the evaluation step, the superiority of the UL relative to baseline aggregation methods is validated to use the standard metrics (i.e., NDCG and diversity). All experiments are performed N times, and N is set as 100. The NDCG and diversity for all aggregation methods are measured as the mean of all experimental measurements.

Furthermore, AU, Avg, MP, SC, AV, BC, CR, MR, AwM and LM are set as the baselines because they correspond to frequently used aggregation methods. With the exception of MR, the predicted ratings of groups are calculated in the existing measurement of the aggregation method. The threshold value of 2 is set in MovieLens for AwM. In the case of the MR, the list of ratings for items is set as a user vector (i.e., $\vec{u} = (r_{u,i_1},\ r_{u,i_2}, \ldots, r_{u,i_n})$), and the average vector for items is calculated by the group (i.e., $\vec{g} = (\overline{r_{i_1}},\ \overline{r_{i_2}}, \ldots, \overline{r_{i_n}})$). This is followed by measuring the distance between users and the average vector by using Euclidean distance. Finally, the individual closest to the average is set as an influential individual in the group, and his/her rating list is set as the predicted ratings.

5.3. Experimental results

The effectiveness of the UL method is evaluated, and it is compared with that of the baseline aggregation methods such that NDCG and diversity can be used. Additionally, a paired student's t-test is performed to verify that the superiority of the UL method

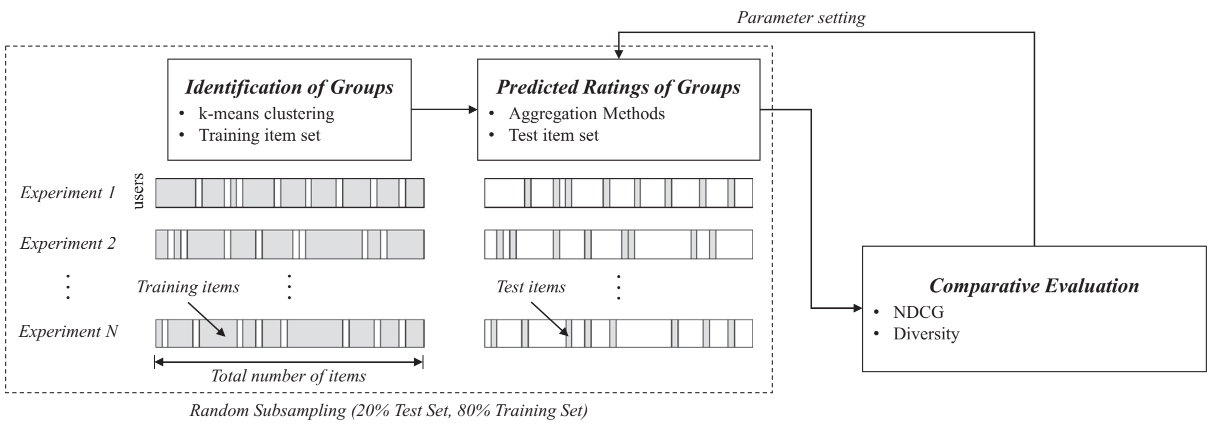

Fig. 2. Evaluation process.

when compared to those of the baselines is statistically significant (p-value $< .05$).

The NDCG is mainly used in the group recommendation, and it corresponds to a metric that is used to evaluate the ranked recommendation results (Baltrunas et al., 2010; Bobadilla et al., 2013; Pessemier et al., 2013). If the top k items are recommended to members in a group G, then the discounted cumulative gain (DCG) and ideal DCG (IDCG) are calculated based on Eqs. (4) and (5), respectively. The NDCG corresponds to the value of DCG divided by IDCG, and it is defined in Eq. (6) as follows:

$$DCG_G = \frac{1}{|G|}\sum_{u\in G}\left(r_{u,p_1} + \sum_{i=2}^{k}\frac{r_{u,p_i}}{\log_2 i}\right) \tag{4}$$

$$IDCG_G = \max(DCG_G) \tag{5}$$

$$NDCG_G = \frac{DCG_G}{IDCG_G} \tag{6}$$

Specifically, $p_1,\ldots, p_n$ denotes the list of the index of ranked items based on the predicted preference rating ($p_{G,i}$) provided by the group. r_{u,p_i} represents the p_ith actual rating given by member u in the group G, and $IDCG_G$ is calculated as the maximum value of DCG. The NDCG is measured for the top k items, and k is set as 5, 10, or 20 in the study.

Table 9 shows the overall effectiveness of the UL and the baselines by using NDCG metrics with the 95% confidence intervals in which the number of group corresponds to 4 (a), 8 (b), 16 (c), 32 (d), 64 (e), and 128 (f). Each symbol adjacent to the baselines indicates the statistical significant improvement over baselines for the UL, according to the paired student's t-test at the 0.05 level (p-value $< .05$). The three combinations of the UL, namely UL_Avg_Dev, UL_AV_Dev, and UL_All, are compared with the baselines. The UL_Avg_Dev corresponds to a combination of the Avg and the Dev, and it is also similar to the Fa2 method. The UL_AV_Dev corresponds to a combination of the AV and the Dev. All elements are considered in the UL_All. The parameter values in the UL, which are α, β, and γ, are determined by measuring the NDCG of all parameter combinations, and a combination of parameters with the highest NDCG value for each group size is selected. The parameter values are indicated below the UL_Avg_Dev, UL_AV_Dev, and UL_All in Table 9.

The Avg, AwM, and LM exhibit high performances in extant studies of group recommendation (Baltrunas et al., 2010; Boratto et al., 2016). Conversely, in the present study, they unexpectedly exhibit a lower performance when compared to those of the other baselines as shown in Table 9. This is because most of the previous studies involved conducting experiments for a small number of group members. Thus, Avg, AwM, and LM methods also exhibit a relatively high performance for a small group size. However, in the present study, an experiment is performed for a large number of group members, and therefore, their performances are significantly lower than those of the other baselines with a large number of group members. Therefore, it is verified that the performance of Avg, AwM, and LM is poor when compared with those of the experiments for a large group size.

Among the all baselines, the MP, MR, and AV methods exhibit a relatively better performance when compared with those of the other baselines. In the case of the MR method, it exhibits high effectiveness for a large number of group members and especially for top rank (NDCG@5). The result shows that the most influential user in a group exerts a high influence on the group. In the case of the AV method, its performance is relatively high when compared with those of others and especially with a small number of groups. This result verified that counting the positive ratings constitutes an important element of the UL. The MP exhibits a good performance in the overall environment. However, a relatively high NDCG value of MP is measured because the rating of the top ranks mostly corresponds to a high rating. Therefore, it is difficult to consider MP as an optimal aggregation method.

Among the three combinations involved in the experiment, the performance of the UL_Avg_Dev is lower than those of the UL_All and the UL_AV_Dev. Even in terms of a comparison with the baselines, the UL_Avg_Dev only exhibits a better performance relative to the Avg, AwM, and LM when the group sizes are large. In contrast, when the group sizes are small, the performance of the UL_Avg_Dev exceeds those of almost all baselines, and the results are statistically significant. The results of the UL_Avg_Dev indicate that Avg is less important when compared with the other elements of the UL. However, when Avg is used with Dev and especially in the case of the small group sizes, they exhibit a complementary relationship that improves the performance.

The performance of UL_AV_Dev corresponds to a further improvement when compared with that of the UL_Avg_Dev. The result verifies that the AV corresponds to a more optimum element when compared with the Avg when the null value in the users-to-ratings matrix is not filled. When compared with the UL_All, the UL_Avg_Dev is slightly poorer in all cases with the exception of the NDCG@10 when the number of group corresponds to 16 and 32. However, it outperforms almost all baselines, and almost all the differences are statistically significant. This result indicates that a high synergy exists when AV is used in conjunction with Dev.

Finally, the full combination, namely UL_All, exhibits the best overall performance with the exception of the NDCG@5 when the number of group corresponds to 8 (the MR represents the opti-

Table 9
Comparative evaluation to measure user satisfaction with ranked list for different group sizes.

Aggregation Methods	*NDCG@k*					
	$k=5$		$k=10$		$k=20$	
(a) The number of groups corresponds to 4						
AU (■)	0.6367 ± 0.0105		0.8282 ± 0.0067		0.9258 ± 0.0024	
Avg (□)	0.1904 ± 0.0233		0.5218 ± 0.0247		0.7729 ± 0.0166	
MP (●)	0.6601 ± 0.0151		0.7412 ± 0.0139		0.8303 ± 0.0056	
SC (○)	0.6236 ± 0.0102		0.8152 ± 0.0071		0.9181 ± 0.0025	
AV (♦)	0.6550 ± 0.0104		0.8409 ± 0.0062		0.9326 ± 0.0021	
BC (◊)	0.6527 ± 0.0107		0.8366 ± 0.0062		0.9294 ± 0.0022	
CR (⌂)	0.6569 ± 0.0100		0.8394 ± 0.0058		0.9287 ± 0.0021	
MR (−)	0.6861 ± 0.0150		0.8038 ± 0.0121		0.8961 ± 0.0061	
AwM (×)	0.0993 ± 0.0135		0.4044 ± 0.0277		0.7121 ± 0.0165	
LM (+)	0.0525 ± 0.0032		0.1236 ± 0.0072		0.3402 ± 0.0155	
UL_Avg_Dev	0.4593 ± 0.0244	□	0.6389 ± 0.0195	□	0.7942 ± 0.0147	□
($\alpha = 0.8$, $\beta = 0$, $\gamma = 0.2$)		×+		×+		×+
UL_AV_Dev	0.6648 ± 0.0105	■□○♦	0.8507 ± 0.0059	■□●○♦	0.9310 ± 0.0025	■□●○♦
($\alpha = 0$, $\beta = 0.4$, $\gamma = 0.6$)		◊⌂×+		◊⌂−×+		◊⌂−×+
UL_All	**0.7137 ± 0.0149**	■□●○♦	**0.8739 ± 0.0068**	■□●○♦	**0.9366 ± 0.0022**	■□●○♦
($\alpha = 0.7$, $\beta = 0.1$, $\gamma = 0.2$)		◊⌂−×+□		◊⌂−×+		◊⌂−×+
(b) The number of groups corresponds to 8						
AU (■)	0.6050 ± 0.0091		0.8137 ± 0.0066		0.9217 ± 0.0032	
Avg (□)	0.2083 ± 0.0171		0.5080 ± 0.0169		0.7385 ± 0.0122	
MP (●)	0.6693 ± 0.0122		0.7881 ± 0.0091		0.8615 ± 0.0058	
SC (○)	0.5929 ± 0.0086		0.7985 ± 0.0068		0.9120 ± 0.0037	
AV (♦)	0.6238 ± 0.0094		0.8349 ± 0.0060		0.9313 ± 0.0024	
BC (◊)	0.6209 ± 0.0087		0.8268 ± 0.0060		0.9278 ± 0.0025	
CR (⌂)	0.6246 ± 0.0088		0.8310 ± 0.0058		0.9278 ± 0.0024	
MR (−)	**0.7156 ± 0.0112**		0.8511 ± 0.0075		0.8968 ± 0.0048	
AwM (×)	0.1494 ± 0.0102		0.4089 ± 0.0199		0.7085 ± 0.0127	
LM (+)	0.1111 ± 0.0084		0.2357 ± 0.0117		0.4145 ± 0.0140	
UL_Avg_Dev	0.4989 ± 0.0176	□	0.6765 ± 0.0118	□	0.8012 ± 0.0107	□
($\alpha = 0.8$, $\beta = 0$, $\gamma = 0.2$)		×+		×+		×+
UL_AV_Dev	0.6267 ± 0.0091	■□○♦	0.8646 ± 0.0064	■□●○♦	0.9328 ± 0.0022	■□●○♦
($\alpha = 0$, $\beta = 0.2$, $\gamma = 0.8$)		◊×+		◊⌂−×+		◊⌂−×+
UL_All	0.7150 ± 0.0155	■□●○♦	**0.8806 ± 0.0064**	■□●○♦	**0.9385 ± 0.0024**	■□●○♦
($\alpha = 0.4$, $\beta = 0.2$, $\gamma = 0.4$)		◊⌂×+		◊⌂−×+		◊⌂−×+
(c) The number of groups corresponds to 16						
AU (■)	0.5718 ± 0.0079		0.7939 ± 0.0069		0.9121 ± 0.0035	
Avg (□)	0.2696 ± 0.0122		0.5545 ± 0.0114		0.8024 ± 0.0079	
MP (●)	0.7047 ± 0.0105		0.8223 ± 0.0059		0.9075 ± 0.0031	
SC (○)	0.5703 ± 0.0079		0.7766 ± 0.0067		0.9027 ± 0.0038	
AV (♦)	0.6020 ± 0.0078		0.8183 ± 0.0062		0.9271 ± 0.0025	
BC (◊)	0.5942 ± 0.0077		0.8060 ± 0.0065		0.9177 ± 0.0034	
CR (⌂)	0.5961 ± 0.0080		0.8096 ± 0.0067		0.9215 ± 0.0029	
MR (−)	0.7257 ± 0.0097		0.8400 ± 0.0057		0.8834 ± 0.0042	
AwM (×)	0.2505 ± 0.0107		0.5054 ± 0.0130		0.7761 ± 0.0074	
LM (+)	0.2184 ± 0.0098		0.4005 ± 0.0107		0.5868 ± 0.0086	
UL_Avg_Dev	0.5719 ± 0.0114	□	0.7550 ± 0.0080	□	0.8886 ± 0.0052	□
($\alpha = 0.8$, $\beta = 0$, $\gamma = 0.2$)		×+		×+		*−×+
UL_AV_Dev	0.7207 ± 0.0093	■□●○♦	**0.8945 ± 0.0048**	■□●○♦	0.9491 ± 0.0018	■□●○♦
($\alpha = 0$, $\beta = 0.2$, $\gamma = 0.8$)		◊⌂×+		◊⌂−×+		◊⌂−×+
UL_All	**0.7478 ± 0.0106**	■□●○♦	0.8938 ± 0.0049	■□●○♦	**0.9493 ± 0.0017**	■□●○♦
($\alpha = 0.5$, $\beta = 0.1$, $\gamma = 0.4$)		◊⌂−×+		◊⌂−×+		◊⌂−×+
(d) The number of groups corresponds to 32						
AU (■)	0.5589 ± 0.0080		0.7738 ± 0.0061		0.9068 ± 0.0033	
Avg (□)	0.3922 ± 0.0090		0.6564 ± 0.0092		0.8749 ± 0.0056	
MP (●)	0.7257 ± 0.0067		0.8602 ± 0.0048		0.9235 ± 0.0019	
SC (○)	0.5622 ± 0.0075		0.7609 ± 0.0060		0.8974 ± 0.0036	
AV (♦)	0.5929 ± 0.0079		0.8035 ± 0.0062		0.9246 ± 0.0031	
BC (◊)	0.5809 ± 0.0080		0.7873 ± 0.0063		0.9031 ± 0.0032	
CR (⌂)	0.5811 ± 0.0085		0.7910 ± 0.0062		0.9149 ± 0.0033	
MR (−)	0.7049 ± 0.0065		0.8142 ± 0.0042		0.8464 ± 0.0030	
AwM (×)	0.3950 ± 0.0073		0.6340 ± 0.0088		0.8628 ± 0.0050	
LM (+)	0.3572 ± 0.0071		0.5533 ± 0.0073		0.7344 ± 0.0051	
UL_Avg_Dev	0.6647 ± 0.0075	■□○♦	0.8361 ± 0.0053	■□○♦	0.9408 ± 0.0024	■□●○♦
($\alpha = 0.7$, $\beta = 0$, $\gamma = 0.3$)		*◊⌂×+		*◊⌂−×+		*◊⌂−×+
UL_AV_Dev	0.7658 ± 0.0067	■□●○♦	**0.9150 ± 0.0038**	■□●○♦	0.9556 ± 0.0013	■□●○♦
($\alpha = 0$, $\beta = 0.2$, $\gamma = 0.8$)		*◊⌂−×+		*◊⌂−×+		*◊⌂−×+
UL_All	**0.7726 ± 0.0070**	■□●○♦	0.9110 ± 0.0042	■□●○♦	**0.9563 ± 0.0015**	■□●○♦
($\alpha = 0.4$, $\beta = 0.1$, $\gamma = 0.5$)		*◊⌂−×+		*◊⌂−×+		*◊⌂−×+

(*continued on next page*)

Table 9 (*continued*)

Aggregation Methods	*NDCG@k*					
	$k=5$		$k=10$		$k=20$	
(e) The number of groups corresponds to 64						
AU (■)	0.5862 ± 0.0069		0.7740 ± 0.0064		0.9001 ± 0.0034	
Avg (□)	0.5212 ± 0.0059		0.7651 ± 0.0055		0.9296 ± 0.0028	
MP (•)	0.7465 ± 0.0054		0.8797 ± 0.0032		0.9283 ± 0.0019	
SC (◦)	0.5969 ± 0.0063		0.7733 ± 0.0061		0.8895 ± 0.0028	
AV (♦)	0.6070 ± 0.0080		0.8086 ± 0.0054		0.9195 ± 0.0026	
BC (◊)	0.5933 ± 0.0072		0.7791 ± 0.0056		0.8931 ± 0.0034	
CR (⌂)	0.5891 ± 0.0076		0.7867 ± 0.0056		0.9095 ± 0.0029	
MR (−)	0.6811 ± 0.0056		0.7735 ± 0.0027		0.7997 ± 0.0027	
AwM (×)	0.5299 ± 0.0053		0.7520 ± 0.0050		0.9227 ± 0.0025	
LM (+)	0.4933 ± 0.0050		0.6882 ± 0.0046		0.8453 ± 0.0028	
UL_Avg_Dev	0.7300 ± 0.0059	■□◦♦	0.8955 ± 0.0034	■□•◦♦	0.9618 ± 0.0016	■□•◦♦
($\alpha = 0.7, \beta = 0, \gamma = 0.3$)		*◊⌂ − × +		*◊⌂ − × +		*◊⌂ − × +
UL_AV_Dev	0.7658 ± 0.0060	■□•◦♦	0.9177 ± 0.0027	■□•◦♦	0.9571 ± 0.0014	■□•◦♦
($\alpha = 0, \beta = 0.2, \gamma = 0.8$)		*◊⌂ − × +		*◊⌂ − × +		*◊⌂ − × +
UL_All	**0.7895 ± 0.0058**	■□•◦♦	**0.9271 ± 0.0025**	■□•◦♦	**0.9623 ± 0.0015**	■□•◦♦
($\alpha = 0.2, \beta = 0.1, \gamma = 0.7$)		*◊⌂ − × +		*◊⌂ − × +		*◊⌂ − × +
(f) The number of groups corresponds to 128						
AU (■)	0.6265 ± 0.0053		0.8035 ± 0.0048		0.9038 ± 0.0025	
Avg (□)	0.6426 ± 0.0041		0.8565 ± 0.0040		0.9577 ± 0.0016	
MP (•)	0.7585 ± 0.0046		0.8842 ± 0.0032		0.9364 ± 0.0014	
SC (◦)	0.6318 ± 0.0054		0.7981 ± 0.0045		0.8943 ± 0.0026	
AV (♦)	0.6481 ± 0.0056		0.8167 ± 0.0046		0.9224 ± 0.0029	
BC (◊)	0.6152 ± 0.0049		0.7712 ± 0.0037		0.8720 ± 0.0031	
CR (⌂)	0.6112 ± 0.0048		0.7867 ± 0.0047		0.8951 ± 0.0030	
MR (−)	0.6466 ± 0.0038		0.7172 ± 0.0024		0.7375 ± 0.0023	
AwM (×)	0.6463 ± 0.0037		0.8474 ± 0.0037		0.9484 ± 0.0015	
LM (+)	0.6140 ± 0.0037		0.7982 ± 0.0032		0.9167 ± 0.0021	
UL_Avg_Dev	0.7448 ± 0.0043	■□◦♦	0.9127 ± 0.0024	■□•◦♦	**0.9579 ± 0.0019**	■•◦♦
($\alpha = 0.6, \beta = 0, \gamma = 0.4$)		◊⌂− × +		◊⌂− × +		◊⌂− × +
UL_AV_Dev	0.7546 ± 0.0043	■□◦♦	0.9081 ± 0.0028	■□•◦♦	0.9465 ± 0.0018	■•◦♦
($\alpha = 0, \beta = 0.2, \gamma = 0.8$)		*◊⌂ − × +		◊⌂− × +		◊⌂−+
UL_All	**0.7921 ± 0.0040**	■□•◦♦	**0.9189 ± 0.0028**	■□•◦♦	0.9553 ± 0.0020	■•◦♦
($\alpha = 0.2, \beta = 0.1, \gamma = 0.7$)		*◊⌂ − × +		◊⌂− × +		◊⌂− × +

mal performance). Furthermore, the UL_All corresponds to a statistically significant improvement when compared with those of all other baselines. The results validate that all three elements in the proposed aggregation method play complementary roles in improving the quality of the group recommendation.

The most important part of the result of Table 9 is that the UL provides satisfactory recommendation results for a large number of group members when compared with existing studies. Thus, the Dev significantly influences the effectiveness of the group recommendation for groups with a large number of members.

Most recommender systems focus on the accuracy of the recommended items. One of the main problems with these systems is that they only recommend popular items to the users. In this case, several users may receive recommended lists containing many similar items. Therefore, we also consider all recommendation lists and measure the degree of differentiation among the recommended items on the lists. A diversity metric is used to evaluate the diversity of the recommended items for the recommender systems (Pessemier et al., 2013). It is calculated by measuring the similarity of movie genre among the recommended items in the movie recommender systems. If i_{genres} denotes the set of genres describing the item i, the similarity of genre between items i and j is based on the Jaccard similarity coefficient and is defined in Eq. (7), as follows:

$$Sim(i,j) = \frac{i_{genres} \cap j_{genres}}{i_{genres} \cup j_{genres}} \tag{7}$$

The similarity of the top k recommended items for group G is calculated by the average of all item pairs' similarities and it is defined in Eq. (8), as follows:

$$Sim_G = \frac{2 \cdot \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} Sim(i,j)}{k \cdot (k-1)} \tag{8}$$

Finally, diversity is calculated by subtracting the average of the similarity of the recommended items for all groups ($\overline{Sim_G}$) and is defined in Eq. (9).

$$Diversity = 1 - \overline{Sim_G} \tag{9}$$

Table 10 shows the diversity metrics with the 95% confidence intervals of the UL and the baselines, and it also indicates statistically significant improvements over the baselines when the number of groups corresponds to 4 (a), 8 (b), 16 (c), 32 (d), 64 (e), and 128 (f).

Among the ULs, the UL_Avg_Dev exhibits a lower diversity value than that of the UL_AV_Dev and the UL_All. Furthermore, it is slightly higher than or almost similar to the baselines. The Avg method also does not exhibit a higher diversity value when compared with those of the other baselines. The result shows that item lists that have a high average do not provide the best recommendation results for the group. However, the UL_Avg_Dev has a higher diversity value than the Avg and, in addition, almost all differences are statistically significant. This result indicates that the Dev compensates for the diversity of the Avg's recommended items.

In some cases, MP has the highest diversity value when the diversity@10 and diversity@20 and when the number of group corresponds to 64, and SC has the highest diversity value when the diversity@20 and when the number of group corresponds to 128. However, in almost all cases, with the exception of the previously mentioned cases, the UL_AV_Dev and the UL_All have relatively

Table 10
Comparative evaluation to measure the diversity of the recommended items list for different group sizes.

Aggregation Methods	*Diversity@k*					
	k = 5		*k* = 10		*k* = 20	
(a) The number of groups corresponds to 4						
AU (■)	0.8243 ± 0.0101		0.8302 ± 0.0054		0.8343 ± 0.0033	
Avg (□)	0.8014 ± 0.0150		0.8082 ± 0.0104		0.8148 ± 0.0058	
MP (•)	0.7979 ± 0.0169		0.8013 ± 0.0114		0.8274 ± 0.0060	
SC (◦)	0.8198 ± 0.0092		0.8268 ± 0.0054		0.8338 ± 0.0030	
AV (♦)	0.8179 ± 0.0098		0.8283 ± 0.0048		0.8295 ± 0.0035	
BC (◊)	0.8203 ± 0.0107		0.8292 ± 0.0053		0.8337 ± 0.0035	
CR (⌂)	0.8286 ± 0.0103		0.8317 ± 0.0049		0.8345 ± 0.0034	
MR (−)	0.7455 ± 0.0153		0.7636 ± 0.0103		0.7691 ± 0.0075	
AwM (×)	0.8231 ± 0.0124		0.8146 ± 0.0097		0.8197 ± 0.0059	
LM (+)	0.8002 ± 0.0160		0.7877 ± 0.0104		0.7835 ± 0.0061	
UL_Avg_Dev	0.8144 ± 0.0126	•−	0.8250 ± 0.0083	□• −×+	0.8194 ± 0.0056	□ −+
UL_AV_Dev	0.8190 ± 0.0106	□• −+	0.8317 ± 0.0057	□•◦ −×+	**0.8384 ± 0.0038**	■□•◦♦ ◊⌂−×+
UL_All	**0.8379 ± 0.0102**	■□•◦♦ ◊−×+	**0.8439 ± 0.0066**	■□•◦♦ ◊⌂−×+	0.8359 ± 0.0046	□•♦ −×+
(b) The number of groups corresponds to 8						
AU (■)	0.8234 ± 0.0061		0.8238 ± 0.0064		0.8347 ± 0.0026	
Avg (□)	0.8169 ± 0.0110		0.8174 ± 0.0060		0.8148 ± 0.0045	
MP (•)	0.7977 ± 0.0144		0.8068 ± 0.0090		0.8302 ± 0.0051	
SC (◦)	0.8182 ± 0.0067		0.8229 ± 0.0045		0.8340 ± 0.0027	
AV (♦)	0.8284 ± 0.0069		0.8354 ± 0.0041		0.8313 ± 0.0029	
BC (◊)	0.8243 ± 0.0069		0.8211 ± 0.0063		0.8354 ± 0.0027	
CR (⌂)	0.8251 ± 0.0061		0.8266 ± 0.0060		0.8364 ± 0.0027	
MR (−)	0.7703 ± 0.0104		0.7962 ± 0.0073		0.8001 ± 0.0053	
AwM (×)	0.8303 ± 0.0110		0.8175 ± 0.0071		0.8188 ± 0.0044	
LM (+)	0.8077 ± 0.0116		0.8119 ± 0.0073		0.8000 ± 0.0049	
UL_Avg_Dev	0.8312 ± 0.0097	□•◦ −+	0.8211 ± 0.0054	• −+	0.8228 ± 0.0045	□ −×+
UL_AV_Dev	0.8375 ± 0.0070	■□•◦♦ ◊⌂−+	**0.8472 ± 0.0044**	■□•◦♦ ◊⌂−×+	**0.8447 ± 0.0027**	■□•◦♦ ◊⌂−×+
UL_All	**0.8395 ± 0.0060**	■□•◦♦ ◊⌂−+	0.8414 ± 0.0044	■□•◦♦ ◊⌂−×+	0.8408 ± 0.0029	■□•◦♦ ◊⌂−×+
(c) The number of groups corresponds to 16						
AU (■)	0.8088 ± 0.0065		0.8173 ± 0.0037		0.8274 ± 0.0025	
Avg (□)	0.8102 ± 0.0069		0.8152 ± 0.0040		0.8221 ± 0.0034	
MP (•)	0.7889 ± 0.0130		0.8061 ± 0.0086		0.8295 ± 0.0040	
SC (◦)	0.8039 ± 0.0067		0.8139 ± 0.0038		0.8272 ± 0.0026	
AV (♦)	0.8087 ± 0.0061		0.8218 ± 0.0035		0.8245 ± 0.0027	
BC (◊)	0.8066 ± 0.0071		0.8188 ± 0.0038		0.8286 ± 0.0027	
CR (⌂)	0.8093 ± 0.0069		0.8193 ± 0.0038		0.8281 ± 0.0025	
MR (−)	0.7885 ± 0.0081		0.7957 ± 0.0043		0.8056 ± 0.0036	
AwM (×)	0.8220 ± 0.0062		0.8196 ± 0.0038		0.8247 ± 0.0033	
LM (+)	0.8031 ± 0.0069		0.8017 ± 0.0047		0.8159 ± 0.0038	
UL_Avg_Dev	0.8206 ± 0.0067	■□•◦♦ ◊⌂−+	0.8190 ± 0.0046	□• −+	0.8249 ± 0.0034	□ −+
UL_AV_Dev	0.8288 ± 0.0046	■□•◦♦ ◊⌂−×+	**0.8354 ± 0.0029**	■□•◦♦ ◊⌂−×+	**0.8346 ± 0.0023**	■□•◦♦ ◊⌂−×+
UL_All	**0.8319 ± 0.0048**	■□•◦♦ ◊⌂−×+	0.8338 ± 0.0031	■□•◦♦ ◊⌂−×+	0.8289 ± 0.0029	□♦ −×+
(d) The number of groups corresponds to 32						
AU (■)	0.7992 ± 0.0051		0.8115 ± 0.0031		0.8217 ± 0.0020	
Avg (□)	0.8088 ± 0.0046		0.8172 ± 0.0039		0.8188 ± 0.0028	
MP (•)	0.7993 ± 0.0103		0.8164 ± 0.0057		**0.8295 ± 0.0032**	
SC (◦)	0.8020 ± 0.0049		0.8117 ± 0.0031		0.8223 ± 0.0019	
AV (♦)	0.8063 ± 0.0044		0.8129 ± 0.0034		0.8237 ± 0.0024	
BC (◊)	0.7990 ± 0.0051		0.8129 ± 0.0032		0.8211 ± 0.0020	
CR (⌂)	0.7996 ± 0.0051		0.8139 ± 0.0031		0.8223 ± 0.0020	
MR (−)	0.7956 ± 0.0050		0.8028 ± 0.0032		0.8115 ± 0.0027	
AwM (×)	0.8092 ± 0.0052		0.8190 ± 0.0036		0.8199 ± 0.0027	
LM (+)	0.8019 ± 0.0049		0.8092 ± 0.0041		0.8159 ± 0.0030	
UL_Avg_Dev	0.8115 ± 0.0058	■•◦ ◊⌂−+	0.8183 ± 0.0033	■◦♦ ◊⌂−+	0.8206 ± 0.0026	□ −+
UL_AV_Dev	0.8144 ± 0.0043	■□•◦♦ ◊⌂−+	**0.8267 ± 0.0023**	■□•◦♦ ◊⌂−×+	0.8272 ± 0.0023	■□◦♦ ◊⌂−×+
UL_All	**0.8177 ± 0.0044**	■□•◦♦ ◊⌂−×+	0.8224 ± 0.0027	■□•◦♦ ◊⌂−×+	0.8213 ± 0.0023	□ −+

(*continued on next page*)

Table 10 (*continued*)

Aggregation Methods	*Diversity@k*					
	k = 5		*k* = 10		*k* = 20	
(e) The number of groups corresponds to 64						
AU (■)	0.7991 ± 0.0042		0.8117 ± 0.0024		0.8218 ± 0.0018	
Avg (□)	0.8092 ± 0.0047		0.8103 ± 0.0029		0.8184 ± 0.0026	
MP (•)	0.8084 ± 0.0080		**0.8225 ± 0.0036**		**0.8268 ± 0.0025**	
SC (◦)	0.7995 ± 0.0044		0.8124 ± 0.0027		0.8237 ± 0.0018	
AV (♦)	0.8048 ± 0.0045		0.8143 ± 0.0032		0.8210 ± 0.0021	
BC (◊)	0.7996 ± 0.0044		0.8126 ± 0.0024		0.8214 ± 0.0018	
CR (⌂)	0.8003 ± 0.0043		0.8133 ± 0.0025		0.8219 ± 0.0017	
MR (−)	0.8036 ± 0.0048		0.8133 ± 0.0029		0.8187 ± 0.0022	
AwM (×)	0.8116 ± 0.0048		0.8125 ± 0.0029		0.8191 ± 0.0024	
LM (+)	0.8070 ± 0.0048		0.8076 ± 0.0030		0.8176 ± 0.0025	
UL_Avg_Dev	0.8069 ± 0.0044	■◦ ◊⌂−	0.8128 ± 0.0028	□ +	0.8193 ± 0.0022	+
UL_AV_Dev	**0.8169 ± 0.0031**	■□•◦♦ ◊⌂−×+	0.8214 ± 0.0023	■□◦♦ ◊⌂− ×+	0.8246 ± 0.0018	■□♦ ◊⌂−×+
UL_All	0.8154 ± 0.0033	■□•◦♦ ◊⌂−+	0.8199 ± 0.0021	■□◦♦ ◊⌂−×+	0.8232 ± 0.0019	■□♦ ◊⌂−×+
(f) The number of groups corresponds to 128						
AU (■)	0.7914 ± 0.0042		0.8045 ± 0.0030		0.8128 ± 0.0029	
Avg (□)	0.7984 ± 0.0048		0.8049 ± 0.0036		0.8099 ± 0.0038	
MP (•)	0.7968 ± 0.0065		0.8093 ± 0.0041		0.8133 ± 0.0035	
SC (◦)	0.7963 ± 0.0045		0.8090 ± 0.0031		**0.8182 ± 0.0026**	
AV (♦)	0.7940 ± 0.0048		0.8038 ± 0.0040		0.8134 ± 0.0030	
BC (◊)	0.7918 ± 0.0042		0.8040 ± 0.0032		0.8116 ± 0.0031	
CR (⌂)	0.7921 ± 0.0041		0.8048 ± 0.0031		0.8126 ± 0.0029	
MR (−)	0.7931 ± 0.0047		0.8041 ± 0.0036		0.8102 ± 0.0033	
AwM (×)	0.8013 ± 0.0046		0.8069 ± 0.0035		0.8115 ± 0.0037	
LM (+)	0.7979 ± 0.0048		0.8052 ± 0.0038		0.8112 ± 0.0036	
UL_Avg_Dev	0.7946 ± 0.0045	■ ◊⌂	0.8038 ± 0.0035		0.8115 ± 0.0033	□ −
UL_AV_Dev	0.8017 ± 0.0046	■♦ ◊⌂−	**0.8118 ± 0.0030**	■□•◦♦ ◊⌂−×+	0.8159 ± 0.0027	■□• ◊⌂− × +
UL_All	**0.8018 ± 0.0042**	■◦♦ ◊⌂−	0.8085 ± 0.0031	■□♦ ◊⌂−×+	0.8138 ± 0.0030	■□ ◊⌂−×+

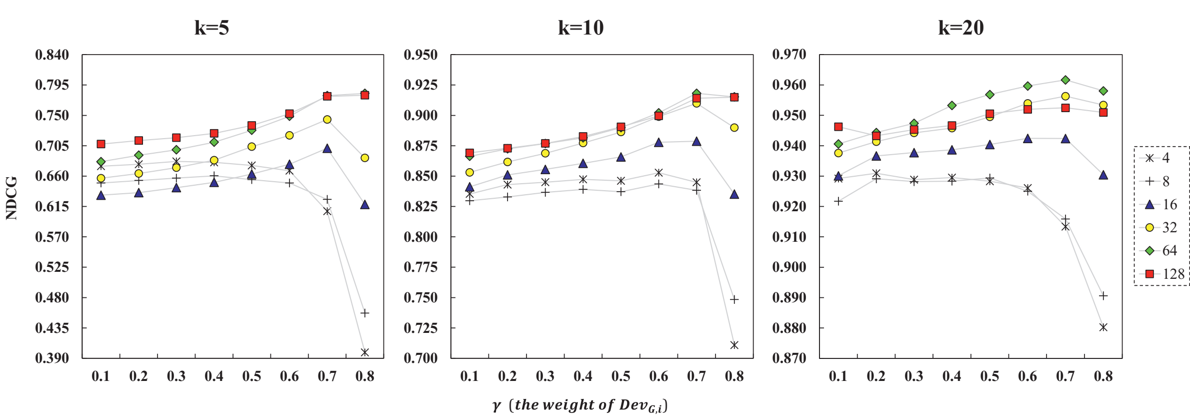

Fig. 3. Comparison of NDCG based on the change in the Dev weight.

higher values with the statistically significant improvements when compared with those of the baselines. They have almost similar diversity values, but the UL_AV_Dev exhibits a slightly higher value when compared with the UL_All. The results indicate that the Avg affects the diversity value of the UL_All worse and the highest synergy for the diversity exists when AV is used together with Dev in the UL.

The influence of Dev in the UL is also evaluated as shown in Fig. 3 that compares NDCG values based on the change in the Dev weight by using MovieLens 100k data set with respect to NDCG@5, NDCG@10, and NDCG@20. The legend to the right of the group indicates the number of groups. Additionally, experiments are conducted only when the weight of all elements in the UL corresponds to or exceeds 0.1. As shown in the Fig. 3., an increase in the Dev weight improves the accuracy of the overall results. Therefore, the results indicate that Dev plays an important role in the UL. Although Dev is important in most cases of the UL, high weights for Dev do not always guarantee a high performance. When the number of groups correspond to 4 and 8, they exhibit difference from the trend of the overall graphs. In these cases, it is difficult to determine whether the importance of Dev is low because the performance difference between the low and high weight of Dev is not high and are almost similar. Furthermore, the results verify that the performance is relatively low when the weight of Dev corresponds to 0.8. This implies that Avg and AV influence the UL and that it is necessary to appropriately mix the three elements of the UL to obtain a high performance.

6. Discussion

In this group recommendation study, we conducted experiments for homogeneous groups that are the sets of members having similar interests. However, in a group recommendation system, heterogeneous groups may need to be considered, depending on the specific environment (Ardissono et al., 2002; Garcia & Sebastia, 2014). For instance, when a family watches the television, it can be regarded as a heterogeneous group because of the difference of their ages and interests. In a group recommendation, recommended results are provided for a homogeneous group or a heterogeneous group depending on whether a group receiving the recommendations is an offline group (Ardissono et al., 2002) or an online group (Boratto, Carta, Chessa, Agelli, & Clemente, 2009), or the system focuses on the users (Garcia & Sebastia, 2014) or the service providers (Boratto et al., 2016).

The recommender system usually provides recommendation results focusing on the user's perspective (Seo et al., 2017). However, there are many costs involved in providing perfectly relevant recommendations to each member in a group. By contrast, from the service provider's point of view, it may be more important to reduce the costs of the recommendation process and provide the reasonable results for the groups at the same time. Consequently, heterogeneous group recommendation studies focus on the group members' satisfaction, while homogeneous group recommendation studies focus on the requirements of the service provider. Therefore, although the former is suitable for group members, the latter is more effective for the service providers.

In case of groups that are gathered on an offline basis (e.g., a family watching TV, or a group of travelers for traveling), the studies focus on the heterogeneous group to address the group recommendations, because they provide the recommended results to the group members who have different interests and perspectives. However, in the case of groups that are gathered on an online basis (e.g., the group members on a specific online system), because they are not being offered recommendations in the same physical space, it would be better to form a group with members who have similar interests and provide recommendations for them. In other words, this case focuses on the homogeneous group.

In this paper, we focus on the homogeneous group as mentioned above. Although existing studies with heterogeneous groups might provide more fair recommendation results to all members in the group (Ardissono et al., 2002; Garcia & Sebastia, 2014), our proposed method is more cost effective than their studies. Furthermore, our study is meaningful in the group recommendation field in that the experiments were conducted on a large number of group members, compared with heterogeneous group recommendation studies that conducted the experiments based on a small number of group members.

7. Conclusion

In this study, an enhanced aggregation method termed as upward leveling is proposed to achieve the aim of a group recommendation. The proposed UL ensures the satisfaction of the maximum number of members in a group with the recommendation results to provide items with an even distribution of their ratings as well as high average, and frequency of rating counts. More specifically, the UL method uses Dev as the most important element, and it is calculated as a proper combination of the Dev, Avg, and AV. The effectiveness of the UL is validated by performing a comparative experiment in which it is compared with the existing aggregation methods based on MovieLens dataset. The results indicate that the UL exhibits superior performances, and it especially provides satisfactory recommendation results in cases involving a large number of group members when compared with extant studies. Furthermore, the results of comparing the performance difference based on the weights of all elements in the UL confirmed that an increase in the weight of Dev leads to an improvement in the performance. Therefore, the results verify that Dev is highly influential and corresponds to the most important element in the UL.

A future study will involve performing experiments by using various data sets that are separate from the MovieLens to verify that the UL exhibits a high performance with respect to most data sets. Specifically, it is necessary to focus on a data set that involves implicit ratings. This is because the UL in the present study is only applicable to explicit ratings. Furthermore, an improved clustering algorithm (relative to the *k*-means algorithm) will be used to detect groups to update the performance of group recommendations. Finally, a future study will consider the optimization of the computation time for group recommendation to efficiently process and handle big data.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2016R1D1A1B03936375).

Appendix

Tables A.1–A.4.

Table A.1
An example of "Simple Computation" methods.

	items				
	i_1	i_2	i_3	i_4	i_5
AU	7	5	11	11	7
Mu	12	6	25	48	12
Avg	2.33	2.5	3.67	3.67	3.5

Table A.2
An example of the MP.

	Items				
	i_1	i_2	i_3	i_4	i_5
MP	3	3	5	4	4

Table A.3
An example of "counting the preferences" methods.

	Items				
	i_1	i_2	i_3	i_4	i_5
SC	3	2	3	3	2
AV	0	0	2	2	1

Table A.4
An example of the MR. Influential person is $\boldsymbol{u}_1$ in Table 1.

	Items				
	i_1	i_2	i_3	i_4	i_5
MR	3	2	5	4	3

References

Agarwal, A., Chakraborty, M., & Chowdary, C. R. (2017). Does order matter? Effect of order in group recommendation. *Expert Systems with Applications, 82*, 115–127.

Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2002). Tailoring the recommendation of tourist information to heterogeneous user groups. In S. Reich, et al. (Eds.), *Hypermedia: Openness, structural awareness, and adaptivity* (pp. 280–295). Berlin: Springer.

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). ACM.

Berkovsky, S., & Freyne, J. (2010). Group-based recipe recommendations: Analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on recommender systems (RecSys'10)* (pp. 111–118). ACM.

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

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., et al. (2010). State-of-the-art in group recommendation and new approaches for automatic identification of groups. In A. Soro, et al. (Eds.), *Information retrieval and mining in distributed environments* (pp. 1–20). Berlin: Springer.

Boratto, L., Carta, S., Chessa, A., Agelli, M., & Clemente, M. L. (2009). Group recommendation with automatic identification of users communities. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology* (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.

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.

Chao, D. L., Balthrop, J., & Forrest, S. (2005). Adaptive radio: Achieving consensus using negative preference. In *Proceedings of the 2005 international ACM SIGGROUP conference on supporting group work (GROUP'05)* (pp. 120–123). ACM.

Christensen, I. A., & Schiaffino, S. (2011). Entertainment recommender systems for group of users. *Expert Systems with Applications, 38*(11), 14127–14135.

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). ACM.

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

Garcia, I., & Sebastia, L. (2014). A negotiation framework for heterogeneous group recommendation. *Expert Systems with Applications, 41*(4), 1245–1261.

Hartigan, J. A., & Wong, M. A. (1979). A *K*-means clustering algorithm. *Applied Statistics, 28*, 100–108.

Jameson, A., & Smyth, B. (2007). Recommendation to groups. In P. Brusilovsky, et al. (Eds.), *The adaptive web* (pp. 596–627). Berlin: Springer.

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

Khoshkangini, R., Pini, M. S., & Rossi, F. (2016). A self-adaptive context-aware group recommender system. In G. Adorni, et al. (Eds.), *AI*IA 2016 advances in artificial intelligence* (pp. 250–265). Cham: Springer.

Lieberman, H., Van Dyke, N., & Vivacqua, A. (1999). Let's browse: A collaborative browsing agent. *Knowledge-Based Systems, 12*(8), 427–431.

Lin, C., Xie, R., Guan, X., Li, L., & Li, T. (2014). Personalized news recommendation via implicit social experts. *Information Sciences, 254*, 1–18.

Liu, Y., Wang, B., Wu, B., Zeng, X., Shi, J., & Zhang, Y. (2016). CoGrec: A community-oriented group recommendation framework. In W. Che, et al. (Eds.), *Social computing* (pp. 258–271). Singapore: Springer.

Mahyar, H., Ghalebi K, E., Morshedi, S. M., Khalili, S., Grosu, R., & Movaghar, A. (2017). Centrality-based group formation in group recommender systems. In *Proceedings of the 26th international conference on world wide web companion (WWW'17 Companion)* (pp. 1187–1196). International World Wide Web Conferences Steering Committee.

Márquez, J. O. Á., & Ziegler, J. (2015). Preference elicitation and negotiation in a group recommender system. In J. Abascal (Ed.), *Human-computer interaction-INTERACT 2015* (pp. 20–37). Cham: Springer.

Márquez, J. O. Á., & Ziegler, J. (2016). Hootle+: A group recommender system supporting preference negotiation. In T. Yuizono, et al. (Eds.), *Collaboration and technology* (pp. 151–166). Cham: Springer.

Masthoff, J. (2015). Group recommender systems: Aggregation, satisfaction and group attributes. In F. Ricci, et al. (Eds.), *Recommender systems handbook* (pp. 743–776). New York: Springer.

McCarthy, J. F. (2002). Pocket restaurantfinder: A situated recommender system for groups. *Workshop on mobile Ad-Hoc communication at the 2002 ACM conference on human factors in computer systems*. ACM.

McCarthy, J., & Anagnost, T. (1998). MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on computer supported cooperative work (CSCW'98)* (pp. 363–372). ACM.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., & Nixon, P. (2006). CATS: A synchronous approach to collaborative group recommendation. In *In proceedings of the Florida artificial intelligence research society conference (FLAIRS'06)* (pp. 86–91). AAAI.

O'connor, M., Cosley, D., Konstan, J. A., & Riedl, J. (2001). PolyLens: A recommender system for groups of users. In W. Prinz, et al. (Eds.), *ECSCW 2001* (pp. 199–218). Dordrecht: Springer.

Pessemier, T., Dooms, S., & Martens, L. (2013). Comparison of group recommendation algorithm. *Multimedia Tools and Applications, 72*(3), 2497–2541.

Quijano-Sánchez, L., Díaz-Agudo, B., & Recio-García, J. A. (2014). Development of a group recommender application in a social network. *Knowledge-Based Systems, 71*, 72–85.

Quijano-Sánchez, L., Recio-García, J. A., & Díaz-Agudo, B. (2015). Modelling Hierarchical Relationships in Group Recommender Systems. In E. Hüllermeier, & M. Minor (Eds.), *Case-based reasoning research and development* (pp. 320–335). Cham: Springer.

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.

Villavicencio, C., Schiaffino, S., Diaz-Pace, J. A., Monteserin, A., Demazeau, Y., & Adam, C. (2016). A mas approach for group recommendation based on negotiation techniques. In Y. Demazeau, et al. (Eds.), *Advances in practical applications of scalable multi-agent systems. The PAAMS collection* (pp. 219–231). Cham: Sprinter.

Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction, 16*(1), 63–82.