

Embedding Implicit User Importance for Group Recommendation

Qing Yang¹, Shengjie Zhou¹, Heyong Li¹ and Jingwei Zhang^{2,3,*}

Abstract: Group recommendations derive from a phenomenon in which people tend to participate in activities together regardless of whether they are online or in reality, which creates real scenarios and promotes the development of group recommendation systems. Different from traditional personalized recommendation methods, which are concerned only with the accuracy of recommendations for individuals, group recommendation is expected to balance the needs of multiple users. Building a proper model for a group of users to improve the quality of a recommended list and to achieve a better recommendation has become a large challenge for group recommendation applications. Existing studies often focus on explicit user characteristics, such as gender, occupation, and social status, to analyze the importance of users for modeling group preferences. However, it is usually difficult to obtain extra user information, especially for ad hoc groups. To this end, we design a novel entropy-based method that extracts users' implicit characteristics from users' historical ratings to obtain the weights of group members. These weights represent user importance so that we can obtain group preferences according to user weights and then model the group decision process to make a recommendation. We evaluate our method for the two metrics of recommendation relevance and overall ratings of recommended items. We compare our method to baselines, and experimental results show that our method achieves a significant improvement in group recommendation performance.

Keywords: Group recommendation, preference aggregation, user importance.

1 Introduction

With the rapid development of recommendation systems, an increasing number of personalized recommendation applications appear in daily life. Mining information that people are interested in from massive data is no longer difficult and instead becomes easy and fast via recommendation systems. Recently, group recommendation has received increasing attention as a new research focus and a novel information service pattern. Moreover, there are many scenarios in the real world for group recommendation. Suppose a group of friends want to see movies together, or a family plans to go on a trip, etc. People

¹ Guangxi Key Laboratory of Automatic Measurement Technology and Instrument, Guilin University of Electronic Technology, Guilin, 541004, China.

² Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, 541004, China.

³ Centre for Applied Informatics, Victoria University, Melbourne, 8001, Australia.

* Corresponding Author: Jingwei Zhang. Email: gtzjw@hotmail.com.

Received: 20 February 2020; Accepted: 26 April 2020.

would like to these activities collectively. Given the potentially numerous options, the group would select a recommendation that is consistent with the preferences of its members.

Different from traditional personalized recommendations, group recommendation treats a group of users as the object of recommendation instead of a single user. When the number of target users increases, the difficulty of recommendation increases greatly because the preferences of a group of users are often inconsistent. A problem with group recommendation is determining how to balance the conflicts of user preferences, that is, how to model group preferences and make better recommendations.

To date, the existing research focuses on how to model group preferences, which has a crucial influence on the effect of group recommendation. On the whole, two methods introduced by Yu et al. [Yu, Zhou, Hao et al. (2006)] are used to model group preferences. The first method creates a joint user profile using a group agent and makes recommendations for this pseudouser. Doubtlessly, this method lacks flexibility and adaptability because it is fixed group-oriented, such that there are no member changes. Due to its narrow application, this method is not commonly used. The second method is the merging method, including the merge preferences method [Jameson and Smyth (2007)] and the merge recommendations method [Baltrunas, Makcinskas and Ricci (2010)], and these two types of methods are compared by Berkovsky et al. [Berkovsky and Freyne (2010)]. Previous research has established that the merge preferences method is used more often.

In this paper, we concentrate on the issue of aggregating preferences to model group preferences. Classic strategies such as the average strategy [Garcia, Pajares, Sebastia et al. (2012); Jameson (2004)] and the least misery strategy [Berry, Fazio, Zhou et al. (2010)] are widely used to achieve effects. However, these strategies ignore user characteristics and only use predefined functions to aggregate user preferences, leading to common results. To simply and efficiently mine user characteristics, we propose a novel entropy-based method that calculates entropy for users, obtaining weights that represent the importance of users in a group. Our method has the following advantages. First, this method only uses user rating data without extra information. This is important because most studies are conducted on nonfixed groups, which makes it more difficult or even impossible to obtain extra information from users. Second, our method utilizes user characteristics to make recommendations and achieves better performance than recommendations based on classic strategies.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 formally defines the group recommendation problem that we studied. In Section 4, we illustrate the methods we proposed to make recommendations for groups and show the algorithms. Section 5 presents an extensive experimental evaluation of movie data, and Section 6 concludes this paper.

2 Related work

A large and growing body of literature has investigated recommendation systems, and most research has attempted to study new algorithms to improve the performance of recommendation systems. Researchers have proposed various methods to find the true intentions of users. Some studies have adopted machine learning methods to learn the implicit characteristics of users and items during multiple iterations of training, and these methods are expected to obtain optimal results [Tran, Pham, Tay et al. (2019); Yin, Ding

and Wang (2019)]. In addition, some researchers have fused extra information of users [Bin, Sun, Cao et al. (2019); Zhu, Wang, Cheng et al. (2018); Yin, Wang, Zheng et al. (2019)] or characteristics of items [Cao, Zhou and Gao (2019)] for better performance. These methods are often costly or require additional information, such as user relationship information, but many achievements have been made.

In contrast to personalized recommendations, in group recommendation studies, to make recommendations better satisfy groups, building a proper group model is essential, and group models are often generated by aggregating individual models.

In an earlier work on group recommendation, O'Connor et al. [O'Connor, Cosley, Konstan et al. (2001)] proposed an algorithm where the missing ratings of users are generated using collaborative filtering, and then score aggregation strategies such as least misery and average strategies are employed to obtain group ratings on items. Subsequently, many works proposed different strategies to generate group models. Among them, the average strategy and least misery strategy are the most popular and are often used as baselines.

The classic average strategy and least misery strategy are designed to satisfy every group member and are often employed to aggregate individual preferences. However, these strategies do not take user characteristics into account. The average strategy treats group members equally, while the least misery strategy always chooses the user with the lowest score as a representative to make group decisions. These strategies and other strategies, such as the most pleasure and most respected person strategies [Masthoff (2011)], are predefined, and a fixed rule is set to aggregate individual preferences. However, it is known that the contributions of group members for modeling group preferences should be different, and the importance of a user changes with the group environment.

In this paper, we introduce the concept of entropy to group recommendation and design an entropy-based method to obtain the importance of group members for group decisions. Some studies have also involved the application of entropy, but most of them have applied entropy to personalized recommendations for different purposes. Saravanan et al. [Saravanan, Mohanraj and Senthilkumar (2019)] also utilized information entropy, and they proposed fuzzy entropy-based deep learning to lower the dimensions of content features as a feature selection method. Additionally, Liu et al. [Liu, Wang and Xu (2018)] proposed a method based on graph entropy to provide weights for two sets of methods of similarity calculations and create the final recommendation lists accordingly.

Unlike the abovementioned approaches, in our research, we use entropy to 'select important users', namely, to weight users. Our method employs only data of user historical ratings and leverages user rating habits to capture the implicit characteristics of users to better model group preferences. Regarding recent studies, the study most closely related to ours is Wang et al. [Wang, Jiang, Sun et al. (2018)], where the authors designed a bidirectional tensor factorization model for group recommendation (BTF-GR) model that learns a function to weight user-item interaction and group-item interaction by capturing the interaction between an individual's interest and the group influence.

3 Problem definition

In this section, we provide a formal statement of the problem of group recommendation and introduce the notation conventions we use in this paper.

Considering a collection U of all users and a collection I of all items, we use $r(u, i)$ to denote the rating of user u for item i . It is generally acknowledged that group recommendation is associated with individual preferences, so we first acquire each member's preferences for items. Due to the sparsity of real rating data, we use a user-based collaborative filtering algorithm to predict the missing ratings.

$$r(u, i) = \frac{\sum_{v \in V} (r(v, i) - \bar{r}_v) \cdot S(u, v)}{\sum |S(u, v)|} \quad (1)$$

Here, V is the set consisting of users who have ratings for item i , and \bar{r}_v is the average rating of user v . $S(u, v)$ denotes the similarity between user u and user v calculated by the Pearson correlation coefficient.

Suppose G is the set consisting of all groups, and all members come from user set U . Given a group of users $g \subset G$ and the item set I , we expect to acquire a package $P \subset I$ that consists of the top- N relevant items to satisfy the group. To obtain the scores of items with regard to a group, we calculate the rating of the group as follows:

$$r(g, i) = F\{r(u, i), u \in g\} \quad (2)$$

where $F\{\cdot\}$ is a function that is designed to transform individual preferences into group preferences; the details are described in the next section. We then retrieve the top- N highest score items, which can be calculated by:

$$Top(g, N) = \arg \max_{i \in I}^N r(g, i) \quad (3)$$

4 Group recommendation methods

As mentioned above, the quality of group recommendation is closely related to the group model. The framework of the generation of group preferences is shown in Fig. 1.

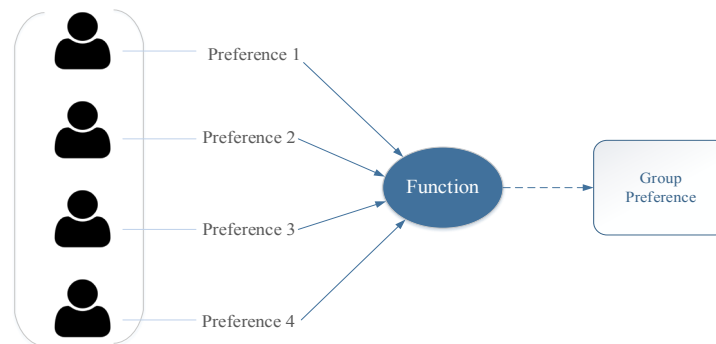


Figure 1: Process of generating group preferences

To fit the group preferences, different studies have aggregated individual preferences using different strategies. Actually, the difference between these strategies lies in the method of weighting users. For instance, in the average strategy, the weight of each user who belongs

to an n-user group is $1/n$, while in the least misery strategy, the weight of the user with the minimum score is 1, and the weights of the other users are 0.

Average (AVG): Given a group of users, the rating of the group for an item is the average of all the members' ratings for the item as follows:

$$r(g, i) = \text{avg} \sum_{u \in g} r(u, i) \quad (4)$$

Least misery (LM): Given a group of users, the rating of the group for an item is the minimum score among all users as follows:

$$r(g, i) = \min_{u \in g} r(u, i) \quad (5)$$

The AVG strategy is popular in many studies because the quality of recommended results under the average strategy is good overall. However, there are certain drawbacks, such as that the AVG strategy may lead a group to obtain items that people neither like nor hate, making users lose interest.

The LM strategy adopts the lowest score among group members to ensure that the recommendation satisfies everyone. Similar to the AVG strategy, this method likely generates recommendations that users dislike because it pays too much attention to user dissatisfaction.

To improve this problem, we try to weight users based on the similarity between users in a group to satisfy more users.

Similarity weighted (SW): Given a group of users, the rating of the group for an item is calculated by:

$$r(g, i) = \sum_{u \in g} w_u \cdot r(u, i) \quad (6)$$

where w_u is calculated based on similarity by:

$$w_u = \frac{\sum_{\substack{v \in g \\ v \neq u}} s(u, v)}{2GS} \quad (7)$$

GS in Eq. (7) is defined as the sum of the similarity between users in the same group.

$$GS = \sum_{\substack{u, v \in g \\ u \neq v}} s(u, v) \quad (8)$$

The SW method is inspired by the thought that a user who is similar to more members in the same group is more representative, and the items this user is interested in will be welcomed by the group in all probability. Therefore, we assign a greater weight to this user. The SW method avoids the case where the recommendation system worries so much about each member that the system makes recommendations that most users are not interested in. However, the SW method may exhibit poor performance with low-inner-similarity groups due to its design principle.

For further optimization, we propose another method based on user characteristics to measure user importance. First, we describe the concept of entropy.

Entropy was first introduced into information theory by C. E. Shannon and became an important concept. Generally, the amount of information is related to the probability distribution of events. In other words, the less likely an event is, the more information it contains. Shannon used information entropy to measure the amount of information.

Currently, entropy is applied in many fields to quantitatively calculate the amount of information, which can be used for feature selection or indicator selection. An index is more suitable as an indicator for evaluating when the index carries more information. The idea behind an entropy-based method is usually the assignment of weights according to the variability of an index.

In this paper, we introduce entropy to group recommendation. It is obvious that users have different habits when rating items. Some users rate items conservatively; for example, a user rates the majority of items he or she experiences as a 3 (five-point scale) and rates items he or she likes as a 4, while never giving any item a 5. In this case, it is difficult to analyze the preferences of this user, and the information this user conveys is considered less useful for modeling group preferences. Our proposed approach hinges upon the key intuition that when a user can output more information, he or she is important to modeling group preferences and should take a higher weight. Now, we illustrate how an entropy-based method is applied to group recommendation.

Entropy-based (EB): Given a group of users, a set of items, and the rating matrix of users with regard to the items, the entropy of users can be calculated by:

$$E_j = -\frac{1}{\ln(n)} \sum_{i=1}^n p_{ij} \ln p_{ij} \quad (9)$$

where E_j denotes the entropy of user j . We regard users in the same group as indexes to judge items simultaneously, calculate the entropy for each user to identify which indexes (users) are more representative for group decisions, and then assign greater weights to them. P_{ij} can be calculated as follows, where i and j represent item i and user j , respectively:

$$P_{ij} = \frac{Y_{ij}}{\sum_{i=1}^n Y_{ij}} \quad (10)$$

In addition, it is necessary to process the rating data before calculation. Y_{ij} in Eq. (9) stands for the value after normalization at row i and column j , and the data are normalized by:

$$Y_{ij} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (11)$$

Having calculated the amount of information users carried using entropy, we then assign weights to users based on the results of the previous step.

According to Eq. (9), we denote the entropy for every user as $\{E_1, E_2, \dots, E_n\}$. Then, the weights of the users can be calculated by:

$$w_n = \frac{1 - E_n}{k - \sum_{n=1}^k E_n} \quad (12)$$

We calculate the group rating in a similar way as Eq. (5), where w_u is replaced with w_n . The algorithm of the proposed EB strategy is given in Algorithm 1.

Algorithm 1. The entropy-based weighted scheme algorithm

Input: a group of users g , set of items I , and rating matrix R

Output: $r(g, i)$

1: Normalize rating matrix R according to Eq. (11)

2: **for** each user $u \in g$ **do**

3: Compute information entropy according to Eqs. (9) and (10)

4: Obtain every member's weight w_n according to Eq. (12)

5: **end for**

6: Compute $r(g, i)$ according to Eq. (6) // update w_u to w_n

7: **return** $r(g, i)$

Compared to the SW method, we expect the EB method to perform stably. This is because the EB method measures user importance according to user characteristics that are relatively stable.

Having defined the method of modeling group preferences, we can then make recommendations for groups. In Algorithm 2, we summarize the recommendation generation algorithm.

Algorithm 2. Recommendation generation algorithm

Input: a group of users g , a set of items I , a rating matrix R for users to items, the size of a recommendation list k , and a candidate list C .

Output: recommendation list L .

1: $L, C = \emptyset$

2: **for** each item $i \in I$ **do**

3: Retrieve rating $r(u, i)$ for each user $u \in g$

4: If $\exists r(u, i) = 0, u \in g$, **do** // the item has not been experienced by any user

5: Continue

6: $C = C \cup \{i\}$ // Add the item to the candidate list

7: **end for**

8: **for** each item $i \in C$ **do**

9: Calculate $r(g, i)$

10: **end for**

11: **while** $|L| < k$ **do**

12: $i^* = \operatorname{argmax}_{i \in C} r(g, i)$

13: $L=L \cup \{i^*\}$

14: $C=C \setminus \{i^*\}$

15: end while

16: return L

5 Experimental results and analysis

In this section, we present an empirical study of our approach on a real dataset to verify the effect of the proposed method. In this paper, recommendation quality refers to two points. First, the quality is related to the relevance of the recommendation list, which is the accuracy of the sequence of items in the recommendation list. Second, the quality can be measured by the overall ratings of recommended items. It is reasonable that the higher a rating is, the better the quality.

The dataset is introduced in Section 5.1, and then we describe the group formation method, evaluation metrics, and results in Sections 5.2, 5.3 and 5.4, respectively. All the algorithms were implemented in Python, and the experiments were run on a computer with an Intel Core CPU @ 3.50 GHz and 8 GB memory.

5.1 Dataset

To the best of our knowledge, there is currently no benchmark dataset for the study of group recommendations for ad hoc groups. Additionally, many studies are devoted to establishing a group model. For this reason, our experiment was carried out on the MovieLens dataset, which includes 6040 users, 3952 items and 1 M integer ratings scaled from 1 to 5. In our experiments, we used only users with a minimum of 100 ratings and items rated by at least 30 users. Then, we ended up with a set of 2945 users and 3952 movies. We used the data for group formation (details are described in Section 5.2), and the data were then divided into a training set and a test set at a 5:5 ratio because our evaluation of the proposed methods depended very strongly on the number of ratings in the test.

5.2 Group formation

In this paper, we conduct research on ad hoc groups where members are changeable, including similar groups that have high inner similarity and random groups. In reality, similar groups usually represent a community with a common taste for a specific activity, while random groups are those meeting unexpectedly, such as people in the same supermarket.

We preliminarily set a group with four members because the design of the similarity weighted strategy requires at least three users in one group. Moreover, the goal of our experiments in this paper is to recommend films to a group, so the size of our group is not large based on realistic experience. We also vary the group size value from 4 to 14 and record the corresponding results for comparative analysis.

In our recommendation scenarios, each user could belong to more than one group. For the formation of a random group, we select a number of users randomly and without any restrictions from user set U . For a similar group, we randomly select a user first and select the next user from users similar to the previous member. We define users whose similarity calculated by the Pearson correlation coefficient with a member is higher than 0.274 as

users similar to the member. We set this threshold because in our dataset, 33% of all user pairs have a similarity higher than 0.274.

5.3 Evaluation metrics

In the experimental evaluation, we evaluate the effectiveness of our group recommendation techniques. First, we use the normalized discounted cumulative gain (NDCG) metric, which has been used in many studies [Baltrunas, Makcinskas and Ricci (2010); Wang, Zhang and Lu (2016); Guo, Tang, Tang et al. (2018); Seo, Kim, Lee et al. (2018)], to estimate the relevance of recommendations. The NDCG metric has the advantage that it not only considers the accuracy of the recommended items but also takes the recommendation order into account.

Given a list of items that are calculated and ranked in descending order by rating via the recommendation system, the DCG of this recommendation list can be calculated by:

$$DCG = \sum_{n=1}^P \frac{2^{r(u,i)} - 1}{\log_2(n+1)} \tag{13}$$

where n represents the position related to the item and the gain is penalized on a logarithmic scale. The DCG takes the rating and order into account but ignores the effects of the number of items and user personal habits on grading. To make the results generated from different methods comparable, the NDCG is proposed to normalize the metric value as follows:

$$NDCG = \frac{DCG}{IDCG} \tag{14}$$

where IDCG is the ideal value of the DCG, which is computed after rearranging these items in optimal order to obtain the ideal DCG. Clearly, a higher NDCG value indicates better performance.

In group recommendation, the computational method is changed to assess the recommendations. In our experiment, we input the list generated by the group recommendation system and obtain an average NDCG value of members in the same group as Wang et al. did [Wang, Zhang and Lu (2016)]. In addition, we obtain the average value of 1000 groups as a result under the same constraint to ensure the validity of the experiment.

The NDCG metric pays more attention to the order of recommended items. Due to the group scenario for recommendation, it is insufficient to evaluate the performance of group recommendation. Therefore, we use the average rating metric, which computes the average rating of recommended items, to measure how many individuals in a group like the recommendation list [Qi, Mamoulis, Pitoura et al. (2016)]. In our experiments, we obtain different recommendation lists based on the abovementioned recommendation approaches and compute the average ratings of group members for items in the recommendation lists. The average rating (AR) score can be computed by:

$$AR = avg \sum_{u=1}^n \sum_{i=1}^k r(u,i) \tag{15}$$

where n and k denote the size of the group and the size of the recommendation list, respectively. For items in the recommendation list of a group, we retrieve real ratings from the test set for each group member. Similar to the NDCG, the average rating is computed

for 1000 groups, and the average is taken. Under this metric, a large score for the AR represents high satisfaction of the group for the recommendation list and further indicates the quality of the recommendation methods.

5.4 Experimental design and results

In this paper, we implement classic and popular group recommendation approaches, the average strategy and least misery strategy, as baselines. We set the size of the recommendation list to 10 and obtain a recommendation list based on different methods, including the EB and SW methods, and we propose baselines under random groups and similar groups. We vary the group size from 4 to 14 and compare the performance of our methods to that of the baselines using the abovementioned NDCG and AR metrics. In addition, we try to combine the EB and SW methods (ES) both with coefficients of 0.5 as a comparison.

Figs. 2 and 3 show the NDCG results obtained according to the AVG, LM, EB, SW and SE methods. As shown in Figs. 2 and 3, regardless of whether groups are established randomly or are established according to similarity, the NDCG of the recommendation list based on the EB method is higher than that of the baselines, which indicates the superiority of this method over other strategies in terms of the quality of ranking. This is expected since the EB algorithm can explore user characteristics from users' historical ratings to better model groups and make group decisions. This is useful regardless of the group environment.

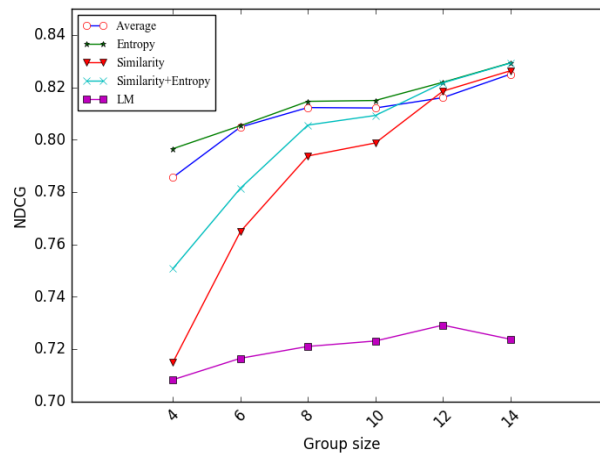


Figure 2: NDCG values for random groups

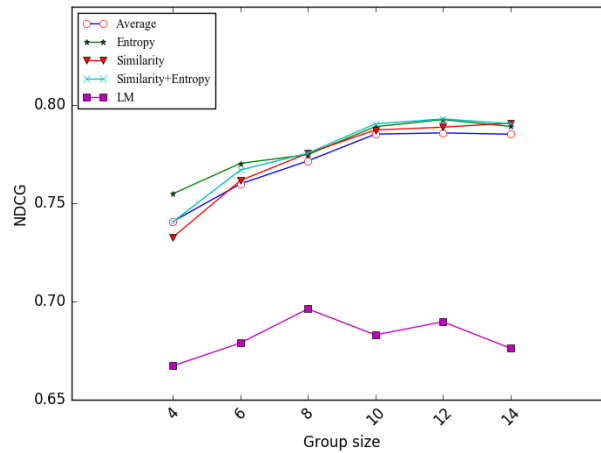


Figure 3: NDCG values for similar groups

We can also observe that the performance of the LM strategy is always the worst because it places too much emphasis on dissatisfaction among users. In addition, we observe that in regard to the random groups, the SW method plays a minimal role. As shown in Fig. 2, for the random groups, the EB, ES, and even AVG methods perform better than the SW method. This can be inferred because the SW strategy relies on user similarity; however, user relationships in random groups are likely to be dissimilar and have more conflict. However, it is worth noting that as the size of the group increases, the SW method works better. We can see in Fig. 2 that when the number of group members increases to 12, the SW method performs better than the AVG method; the SW method has little effect when group members are dissimilar, while in a similar group, as shown in Fig. 3, the SW method plays an important role and achieves better performance than the baselines most of the time. This is expected due to its design principles.

We calculate the average ratings of the recommendation lists obtained by different strategies. Figs. 4 and 5 show the AR results under random groups and similar groups. The results are consistent with the NDCG metrics. We observe that the EB strategy always performs better than the baselines. As shown in Fig. 4, the average rating based on the SW strategy under the random groups does not perform well with a small group, but it performs better than the AVG strategy when the group size increases. According to the results in Fig. 5, under similar groups, the SW strategy significantly improves in performance with increasing group size and even obviously exceeds the AVG strategy when the group size is larger than 8. This indicates that the similarity factor plays a large role in similar groups. Similarly, it is obvious that the recommendation list based on the LM strategy always scores the lowest due to its design principle. In addition, it is interesting that the ES strategy sometimes performs better than the EB and SW strategies, even though it is just a simple combination of these two methods, and we plan to carry out more research on this aspect in future works.

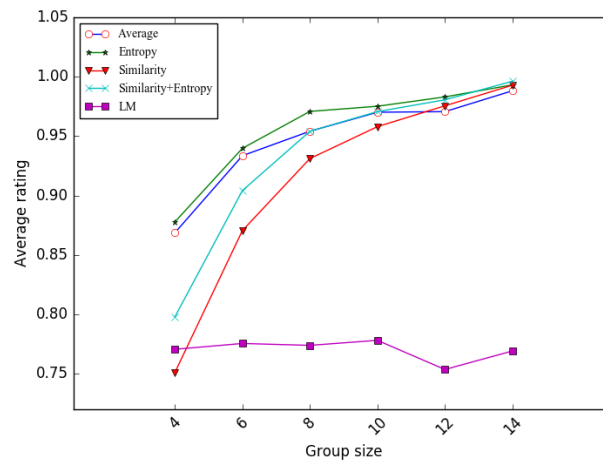


Figure 4: Average ratings for random groups

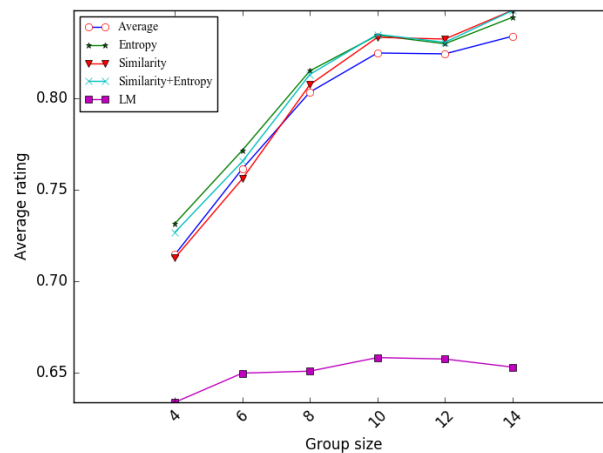


Figure 5: Average ratings for similar groups

6 Conclusions

In this paper, we conduct research on the problem of the quality of group recommendation with regard to the relevance of the recommended items and the overall ratings of the recommended items. We propose two methods for aggregating individual preferences to improve the performance of a recommendation system and verify the effectiveness of these methods by the state-of-the-art NDCG and AR metrics. Experimental results on real datasets show that the recommendations based on the EB strategy are superior to those under baseline approaches. The SW strategy works better when group members have similar tastes and plays an important role with large-size groups. In other words, the methods we proposed maintain a higher-quality recommendation list in terms of the accuracy of the ranking and overall ratings of items.

Funding Statement: This study is funded by the National Natural Science Foundation of China (Nos. 61862013, 61662015, U1811264, and U1711263), Guangxi Natural Science Foundation of China (Nos. 2018GXNSFAA281199 and 2017GXNSFAA198035), Guangxi Key Laboratory of Automatic Measurement Technology and Instrument (No. YQ19109) and Guangxi Key Laboratory of Trusted Software (No. kx201915).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- Baltrunas, L.; Makcinskas, T.; Ricci, F.** (2010): Group recommendations with rank aggregation and collaborative filtering. *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 119-126.
- Berkovsky, S.; Freyne, J.** (2010): Group-based recipe recommendations: analysis of data aggregation strategies. *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 111-118.
- Berry, S.; Fazio, S.; Zhou, Y.; Scott, B.; Revilla, L. F.** (2010): Netflix recommendations for groups. *Proceedings of the American Society for Information Science and Technology*, vol. 47, no. 1, pp. 1-3.
- Bin, S.; Sun, G.; Cao, N.; Qiu, J.; Zheng, Z. et al.** (2019): Collaborative filtering recommendation algorithm based on multi-relationship social network. *Computers, Materials and Continua*, vol. 60, no. 2, pp. 659-674.
- Cao, M.; Zhou, S.; Gao, H.** (2019): A recommendation approach based on product attribute reviews: improved collaborative filtering considering the sentiment polarity. *Intelligent Automation and Soft Computing*, vol. 25, no. 3, pp. 595-604.
- Garcia, I.; Pajares, S.; Sebastia, L.; Onaindia, E.** (2012): Preference elicitation techniques for group recommender system. *Information Sciences*, vol. 189, pp. 155-175.
- Guo, Z.; Tang, C.; Tang, H.; Fu, Y.; Niu, W.** (2018): A novel group recommendation mechanism from the perspective of preference distribution. *IEEE Access*, vol. 6, pp. 5865-5878.
- Jameson, A.** (2004): More than the sum of its members: challenges for group recommender system. *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 48-54.
- Jameson, A.; Smyth, B.** (2007): Recommendation to groups. *The Adaptive Web: Methods and Strategies of Web Personalization*, vol. 4321, pp. 596-627.
- Liu, S.; Wang, B.; Xu, M.** (2018): SERGE: successive event recommendation based on graph entropy for event-based social networks. *IEEE Access*, vol. 6, pp. 3020-3030.
- Masthoff, J.** (2011): *Group Recommender Systems: Combining Individual Models. Recommender Systems Handbook*. Springer, Boston.
- O'Connor, M.; Cosley, D.; Konstan, J. A.; Riedl, J.** (2001): PolyLens: a recommender system for groups of users. *Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work*, pp. 199-218.
- Qi, S.; Mamoulis, N.; Pitoura, E.; Tsaparas, P.** (2016): Recommending packages to

groups. *IEEE 16th International Conference on Data Mining*, pp. 449-458.

Saravanan, B., Mohanraj, V.; Senthilkumar, J. (2019): A fuzzy entropy technique for dimensionality reduction in recommender systems using deep learning. *Soft Computing*, vol. 23, no. 8, pp. 2575-2583.

Seo, Y. D.; Kim, Y. G.; Lee, E.; Seol, K.; Baik, D. K. (2018): An enhanced aggregation method considering deviations for a group recommendation. *Expert Systems with Applications*, vol. 93, pp. 299-312.

Tran, L. V.; Pham, T. N.; Tay, Y.; Liu, Y.; Cong, G. et al. (2019): Interact and decide: medley of sub-attention networks for effective group recommendation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 255-264.

Wang, J.; Jiang, Y.; Sun, J.; Liu, Y.; Liu, X. (2018): Group recommendation based on a bidirectional tensor factorization model. *World Wide Web*, vol. 21, no. 4, pp. 961-984.

Wang, W.; Zhang, G.; Lu, J. (2016): Member contribution-based group recommender system. *Decision Support Systems*, vol. 87, pp. 80-93.

Yin, C.; Ding, S.; Wang, J. (2019): Mobile marketing recommendation method based on user location feedback. *Human-Centric Computing and Information Sciences*, vol. 9, no. 14, pp. 1-17.

Yin, H.; Wang, Q.; Zheng, K.; Li, Z.; Yang, J. et al. (2019): Social influence-based group representation learning for group recommendation. *International Conference on Data Engineering*, pp. 566-577.

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*, vol. 16, no. 1, pp. 63-82.

Zhu, Q.; Wang, S.; Cheng, B.; Sun, Q.; Yang, F. et al. (2018): Context-aware group recommendation for point-of-interests. *IEEE Access*, vol. 6, pp. 12129-12144.