

A Group Recommendation Algorithm Based on Group Members' Weights and Project Popularity

Min Li^{1,a}, Chunming Wu^{1,b,}, and Li Ye^{1,c}*

*College of Computer and Information Science Southwest University No.2, Tiansheng Rd., Beibei, Chongqing 400715, P.R.China
^alm044221@email.swu.edu.cn, ^bspringsun@swu.edu.cn, ^c372166190@qq.com*

Keywords: Group recommendation, Collaborative Filtering, Members' Weights, Project Popularity.

Abstract

With the rapid development of many types of social networks, the number of people involved in group activities is increasing, and thus group recommendation systems have been widely studied. This paper proposes a collaborative filtering algorithm to mine groups' interests. In view of the different roles played by different users in the group, the algorithm combines the members' weights factor. And because of the differences in the degree of concern among groups and members for the various categories, the weighted mean similarity method is used. In order to solve the problem of a groups' narrow field of view of the existing algorithms and to more effectively mine unpopular items, the algorithm considers project popularity to improve the novelty of the recommendations. Experiments on the MovieLens 1M GroupLens data set show that the algorithm can effectively improve the accuracy of the recommendations. And to a certain extent, it improves the novelty of the recommendations.

1 Introduction

With the exponential growth of information in the Internet era, providing users with more accurate personalized service is an increasingly important research field. Therefore, an intelligent way, which can automatically provide a specific recommendation service for the users with their specific needs is expected. Recommendation systems emerged as the required by the times^[1]. Depending on the different service objects, the recommendation system can be divided into two categories: an individual user recommendation system or a group user recommendation system. The former studies how to provide personalized recommendations for individual users. While the latter studies how to provide differentiated referral services for different groups, making each member of the group as satisfied as possible. For example, according to different interests, a family can be recommended a group of travel information, while another family can be recommended information about films. Early researches focused predominately on personal recommendations. Some research results have been applied in the related fields such as tourism^[2], social news^[3], TV programs^[4], etc.. In recent years, group recommendations have been widely studied. Many scholars focused on existing and non-existing group recommendations, user interest modeling, group rating prediction accuracy and recommendation accuracy and other

issues. They have put forward some recommendation algorithms and designed some experimental prototype system^[5,6,7,8,9,10]. But much of the work does not take into consideration in the differences in the level of concern of different groups for each category of project, as well as differences in the level of concern among members of the same group for each category of item. This results in the similarity calculation between groups and users is not accurate enough. Some systems excessive pursuit of the accuracy of the recommendations causes a narrow vision of the group, reducing the recommended coverage and novelty. And some work considers that the accuracy of the prediction group score is the standard to measure the quality of the system. But the aim of the group recommendation system is to provide accurate and novel recommendation services for the community. This evaluation system is clearly biased. In this paper, a group recommendation algorithm is proposed based on group members' weights and project popularity. The introduction of the members Group weights allows for more accurate calculation of group preferences, making the recommendation more accurate. Project popularity can effectively uncover popular projects, improve the recommendation coverage and reduce the popularity of recommended items.

2 Related work

Collaborative filtering. The collaborative filtering recommendation algorithm was the first proposed algorithm that provided the most thorough research in the recommendation system. According to whether to join the model, the collaborative filtering recommendation algorithm can be divided into two categories: heuristic (also known as memory-based) and model-based. According to the purpose and function, it can be divided into two categories: score prediction and TopN recommendation. Among them, the most classic memory-based collaborative filtering algorithm mainly includes project-based^[11] and user-based collaborative filtering.

Group user recommendation. The group recommendation algorithm integrates the preferences of all the members of the group and provides the recommended service for the group. Over the past few years, some well-known group recommendation systems have been developed and applied in practice^[5,6,7]. However, more accurate personalized recommendation service can't be provided. More recently, many scholars have conducted in-depth research on how to improve the effect of group recommendation, and put forward many related algorithms. Chao et al.^[8] found the nearest neighbors of the group by computing groups and users of a similarity matrix, so as to predict the behavior of the group according to

the behavior of the nearest neighbors. But the method does not consider the role of the user in the group, which affects the quality of the recommendations. Ghazarian et al.^[9] proposed a collaborative filtering algorithm based on the user and the item, which used a support vector machine model to calculate the similarity of the project. This method effectively solved the data sparsity problem, but the pursuit of the accuracy of the predictive score, the quality of the recommended items was ignored. The effect of the recommended service was not good. Ortega et al.^[10] proposed a matrix decomposition method to build a group recommendation system, which had three decomposition strategies. But the matrix decomposition method had the disadvantage that the recommended list is less interpretable. In general, group recommendation is still in the exploratory stage. There are still many problems to be studied and improved. This paper proposes a group recommendation algorithm based on group members' weights and project popularity. The advantage of this algorithm is that it considers the influence of the weight of the members on the accuracy of the recommendation and the influence of the project popularity regarding novelty.

3 Group Recommendation Method

In view of the shortcomings of the existing algorithms, this paper propose a group recommendation algorithm based on group members' weights and project popularity. The membership weight (MW) factor is introduced into the calculation of behavior similarity to enhance the accuracy of the recommendations, and a project popularity(PP) parameter is introduced to improve coverage, novelty and surprise of the recommendations. The algorithm mainly includes the following steps: First, the user is clustered and grouped based on the similarity of user behavior. Then it calculates the similarity between the group and user behavior. Finally, the group score is predicted based on the behavior of the K users most similar to the behavior of the group, and the N items with the highest score is recommended to the group.

3.1 Group division

As "social animals", people are not grouped by chance. Therefore, it's necessary to group people according to the user's behavioral similarity. Due to the sparsity of user ratings, the deviation of user's behavior similarity is large. In real life, goods will usually have their own categories, and the user's preferences to the items to a certain extent reflect the user's interests. Therefore, user item-class preferences can be used to calculate the similarity of user behavior compensating for the lack of data sparsity.

Definition 1 User item-class preference^[12] matrix. Let the item space be $I = \{i_1, i_2, \dots, i_n\}$, u is one user in user space. $I_u = \{i \in I | R_{u,i} \neq \bullet\}$ denotes the item set that has been scored by u (\bullet represents a null value). $C = \{c_1, c_2, \dots, c_l\}$ represents an item-class set,

$\forall c \in C \rightarrow p_{u,c} = \#(c \cap I_u) / \#I_u$ is the item-class preference value of u to c ($\#$ represents the set of potentials). $P(m, I)$ is the user item-class preference matrix. Using $P(m, I)$, K-means^[13] algorithm is used to divide users into k categories that are subdivided into a number of groups, each group of 2-8 members.

3.2 Group and user behavior similarity

Member's weight. In the group decision-making process, each member's idea will have a different influence on the group's choice. Therefore, for the similarity calculation between the group and the user behavior, the MW factor needs to be integrated. For the calculation of membership weight, the this paper considers the following questions: 1) For a large number of items, a group may only be interested in a specific number of categories. And the degree of emphasis on the different classes may be different. 2) For each of these categories, the degree of concern of each member is also different. 3) Regarding the practical application, each user's score times (number of transactions) are often not the same. Some members who had fewer score times is not representative in a group. Therefore, first, this paper calculates the member's mean similarity within every class for any group, as the interest of each member in each class. And then calculates the preferences of each item-class, used to weight the interest of each member in each class. So that, the weighted item class mean similarity (WICMS) can be obtained. Finally, the member rating frequency weight (RFW) is introduced. Then the formula of MW is $MW = WICMS \cdot RFW$.

Definition 2 Mean similarity within every class. Let $G = \{g_1, g_2, \dots, g_t\}$ represent a group, $\forall c \in C \wedge c = \{i_1, i_2, \dots, i_n\} \rightarrow R_{g,c}$ is the rating matrix of G to c . $\forall g \in G \rightarrow R_{g,c}$ represents the rating vector of the member g in G to c . $\bar{R}_{g,c} = (\sum_{g \in G} R_{g,c}) / t$ represents the mean rating vector of the G to c . Using Pearson correlation coefficient^[13], calculate the similarity relation between $R_{g,c}$ and $\bar{R}_{g,c}$. Then the mean similarity within every class of g is:

$$\forall g \in G \rightarrow sim(R_{g,c}, \bar{R}_{g,c}) = \frac{\sum_{i \in c} (r_{g,i} - \bar{r}_g) \cdot (\bar{r}_{g,i} - \bar{r}_c)}{\sqrt{\sum_{i \in c} (r_{g,i} - \bar{r}_g)^2} \sqrt{\sum_{i \in c} (\bar{r}_{g,i} - \bar{r}_c)^2}} \quad (1)$$

Where $r_{g,i}$ and $\bar{r}_{g,i}$ respectively represent the score and mean score of member g and group G to item i in item-class c . \bar{r}_g and \bar{r}_c respectively represent the mean score of g and G to c . Table 1 shows $R_{g,c}$.

Definition 3 Group item-class preference. Let $\forall c \in C \rightarrow p_{g,c} = \#(c \cap I_g) / \#I_g$ represent the preference of g in G to c , $P_g(t, I)$ represents the preference matrix of G to the item-class set C . Then $\bar{p}_{g,c} = (\sum_{g \in G} p_{g,c}) / t$ represents the preference of G to c . Where t represents the number of item-classes.

$$WICMS_g = \sum_{c \in C} sim(R_{g,c}, \bar{R}_{g,c}) \cdot \bar{p}_{g,c} \quad (2)$$

	i_1	i_2	\dots	i_n		c_1	c_2	\dots	c_l
g_1	$r_{1,1}$	$r_{1,2}$	\dots	\dots	g_1	$p_{1,1}$	$p_{1,2}$	\dots	$p_{1,l}$
g_2	$r_{2,1}$	\dots	\dots	$r_{2,n}$	g_2	$p_{2,1}$	$p_{2,2}$	\dots	$p_{2,l}$
\vdots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots	\vdots	\dots	\vdots
g_t	$r_{t,1}$	$r_{t,2}$	\dots	$r_{t,n}$	g_t	$p_{t,1}$	$p_{t,2}$	\dots	$p_{t,l}$

Table 1 $R_{g,c}(t, n)$

Table 2 $P_c(t, l)$

Definition 4 RFW. For any group G , the rating frequency weight of g in G is:

$$\forall g \in G \rightarrow RFW_g = \# I_g / \sum_{u \in G} \# I_u \quad (3)$$

Where the greater the ratio of the number of g's rates to the number of G's rates, means the greater the rating frequency weight of g. And g is the greater role in the process of group decision-making. The MW of g is:

$$MW_g = WICMS_g \cdot RFW_g \quad (4)$$

Project popularity parameter. The higher the popularity of the project, which means that the higher exposure of the project, suggests there to be more user feedback. But the project does not necessarily reflect the user's personalized preferences. Instead, users' feedback to items of low popularity can reflect the users' personalized preferences. In order to avoid the excessive pursuit for the accuracy of the recommendations which may cause a narrow vision of the entire group, project popularity weights are introduced to improve the coverage and novelty by reducing the average popularity of recommended projects.

Definition 5 pp parameter.

$$PP(i) = avg_pop / item_pop(i) \quad (5)$$

represents the project popularity parameter of item i . Where avg_pop represents average popularity of all projects. And $item_pop(i)$ represents project popularity of i .

Group and user similarity. Based on the above analysis, this paper proposes the following definition of group and user similarity as a fusion of member weights and project popularity parameters(GUSFMP).

Definition 6 GUSFMP.

$$GUSFMP(G, u) = \left\{ \begin{array}{l} \frac{1}{\sum_{i \in I_{G,u}} PP(i)} \cdot \sum_{i \in I_{G,u}} \frac{\sum_{g \in G \wedge r_{g,i} = r_{u,i}} MW_g}{\sum_{g \in G \wedge r_{g,i} \neq \bullet} MW_g} \cdot PP(i) \Leftrightarrow I_{G,u} \neq \emptyset \wedge u \notin G \\ \bullet \Leftrightarrow I_{G,u} = \emptyset \vee u \in G \end{array} \right\} \quad (6)$$

Where $I_{G,u}$ represents a collection of projects rated by groups and users together.

Then so as to reflect the function of the member weights and project popularity parameters separately, two formulas are given to verify the validity of these two weights.

1) The similarity of the groups and users with only members' weights(GUSFM):

$$GUSFM(G, u) = \left\{ \begin{array}{l} \frac{1}{\# I_{G,u}} \cdot \sum_{i \in I_{G,u}} \frac{\sum_{g \in G \wedge r_{g,i} = r_{u,i}} MW_g}{\sum_{g \in G \wedge r_{g,i} \neq \bullet} MW_g} \Leftrightarrow I_{G,u} \neq \emptyset \wedge u \notin G \\ \bullet \Leftrightarrow I_{G,u} = \emptyset \vee u \in G \end{array} \right\} \quad (7)$$

2) The similarity of the groups and users with only project popularity(GUSFP):

$$GUSFP(G, u) = \left\{ \begin{array}{l} \frac{1}{\sum_{i \in I_{G,u}} PP(i)} \cdot \sum_{i \in I_{G,u}} \frac{\# \{g \in G \mid r_{g,i} = r_{u,i}\}}{\# \{g \in G \mid r_{g,i} \neq \bullet\}} \cdot PP(i) \Leftrightarrow I_{G,u} \neq \emptyset \wedge u \notin G \\ \bullet \Leftrightarrow I_{G,u} = \emptyset \vee u \in G \end{array} \right\} \quad (8)$$

3.3 Group score prediction

To predict the group score, we must first define the group's neighbor users. According to the group and user similarity in the 3.2 section, K maximum similarity of users are as a group of nearest neighbor users. Formally:

$$K_G \subseteq U \wedge \# K_G = k \wedge \forall x \in K_G, \forall y \in (U - K_G) : GUSFMP(G, x) \geq GUSFMP(G, y)$$

Let $P_{G,i} = \{u \in K_G \mid r_{u,i} \neq \bullet\}$ represents the set of neighbors which have rated item i , The predicted G 's score for i is:

$$P_{G,i} = \left\{ \begin{array}{l} \frac{1}{\sum_{u \in P_{G,i}} GUSFMP(G, u)} \sum_{u \in P_{G,i}} GUSFMP(G, u) \cdot r_{u,i} \Leftrightarrow P_{G,i} \neq \emptyset \\ \bullet \Leftrightarrow P_{G,i} = \emptyset \end{array} \right\} \quad (9)$$

3.4 Recommendation

First, Let $X_G \subseteq (I - I_G) \wedge \forall i \in X_G : p_{G,i} \neq \bullet$ represents the collection of items can be recommended to the group G . Let Y_G represent he top N items with the highest predicted scores. Formally:

$$Y_G \subseteq X_G, \# Y_G = \min(N, \# X_G), \forall x \in Y_G, \forall y \in (X_G - Y_G) : p_{G,x} \geq p_{G,y}$$

3.5 Algorithm description

Based on the above analysis, the algorithm description of GUSFMP is in Table 3.

Algorithm: A Group Recommendation Algorithm Based on Group Members' Weights and Project Popularity

Input: List of items and their categories, User-item rating matrix, Proportion of training set, Number of tested groups, Group size, k, N, θ .

Output: Top N recommendations for each group.

Step 1 Divide users into K categories using Table 1 $P(m, l)$ and K-means algorithm, then subdivided into a number of groups.

Step 2 Calculate the avg_pop ;

Step 3 For each item i {

Calculate the $item_pop(i)$;

Calculate the $PP(i)$ according formula (5);

Step 4 For each group G {

Calculate the $WICMS_g, RFW_g, MW_g$ according to the formula (2)-(4) for each g ;

The $GUSFMP(G, u)$ is calculated according to the formula (6) for each user u ;

Step 5 Find the K_G for each group G based on k and similarity matrix;

Step 6 For each group G

For each item i {

Find the $P_{G,i}$ about i based on K_G ;

The $p_{G,i}$ is calculated according to the formula (9); }

Step 7 Get Y_G for each group G based on N and predictive scoring matrix.

Table 3: The algorithm description of GUSFMP

4 Experiment

4.1 Data sets and metrics

This article selects the Movielens 1M test data set from the MovieLens (<http://movielens.umn.edu>) site. The data set contains a total of 1,000,209 scoring records from 6,040 users for 3,706 items, with a sparse rating of $1-1000209 / (6040 * 3706) = 95.53\%$. Where, there are 207 items rated more than 1000 times. The total number of feedback was 305,373. And there are 3,499 items rated more than 1000 times. The total number of feedback was 694,836. This indicates that the low popularity projects have a huge number of overall feedback. Fully mining these unpopular projects can improve business performance, but also can give users better service. The purpose of this paper is to ensure improved accuracy while maximizing the recommended coverage and novelty.

Therefore, five performance indexes are selected to evaluate the effectiveness of the algorithm. The indexes and their

formulas are as follows: $accuracy = \frac{\#(TP_G \cup TN_G)}{\#(TP_G \cup FP_G \cup FN_G \cup TN_G)}$,

$precision = \frac{\# TP_G}{\#(TP_G \cup FP_G)}$, $recall = \frac{\# TP_G}{\#(TP_G \cup FN_G)}$,

$coverage = \frac{\# R_G}{\# I}$, $prevalence = \frac{\sum_{i \in R_G} item_pop(i)}{N}$

where $TP_G = \{i \in R_G \mid \exists g \in G \text{ such that } r_{g,i} \neq \bullet \text{ and } \forall u \in G \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\}$ represents True Positive, positive samples predicted by the model as positive;

$TN_G = \{i \in R_G \mid \forall g \in G \rightarrow r_{g,i} = \bullet \text{ and } \exists u \in G \hat{r}_{u,i} \neq \bullet \text{ such that } \hat{r}_{u,i} < \theta\}$ represents True Negative, negative samples predicted by the model as negative;

$FP_G = \{i \in R_G \mid \exists g \in G \text{ such that } r_{g,i} \neq \bullet \text{ and } \exists u \in G \hat{r}_{u,i} \neq \bullet \text{ such that } \hat{r}_{u,i} < \theta\}$ represents False Positive, negative samples predicted by the model as positive;

$FN_G = \{i \in R_G \mid \forall g \in G \rightarrow r_{g,i} = \bullet \text{ and } \forall u \in G \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\}$ represents False Negative, positive samples predicted by the model as negative. Where R_G represents the set of items to be recommended to the group G . $\hat{r}_{u,i}$ represents the test-rating of the user u to the item i . And θ is a threshold to measure whether a user likes or dislikes an item.

Table 4 shows the parameters.

parameter	value
Train set percent	65%
Number of test group	1200
Group size	2-8
Small group size	2-4

Large group size	5-8
Number of neighbors	200
θ	4,5/1,2,3

Table 4 Main parameters of the experiment

4.2 Precision/Recall comparison

Figure 1 shows the results of the system Precision / Recall comparison for small group (a) and large group (b) respectively. Comparing the two graphs, we can see that, on Precision, the small group result range is [0.6531-0.6906], while the large group result range is [0.5926-0.6819]. In the Recall, the small group result range is [0.0082- 0.0671], while the large group result range is [0.0056-0.0551]. Which indicates that the small group has better recommendation effect. At the same time, it can be found that in small or large group recommendation, when the recommended number is less than 7, the quality fluctuation of each algorithm is significant. But when the recommended number is more than 7, the GUSFMP algorithm proposed in this paper is significantly more accurate than the other three algorithms.

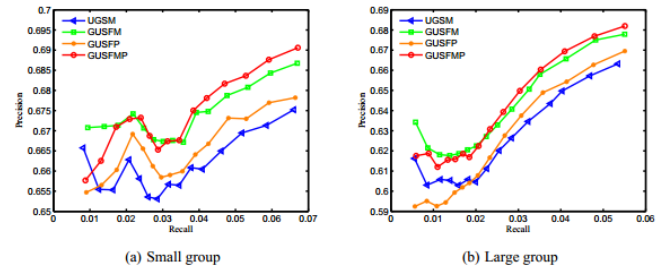


Fig. 1: Precision/Recall comparison

and white so take special note of any colours you have used in diagrams.

4.3 Accuracy/Coverage comparison

Figure 2 shows the results of the system Accuracy/Coverage comparison for the small population (a) and the large population (b). Comparing the two graphs, on Accuracy, the small group result range is [0.7561-0.7803], while the large group result range is [0.7868-0.8062]. For Coverage, the small group result range is [0.1262-0.7331], while the large group result range is [0.1565-0.7576]. This indicates that the large group has a better recommendation effect. At the same time, it can be found that in small or large group recommendations, the GUSFMP algorithm proposed in this paper has a significantly better effect than the other three algorithms.

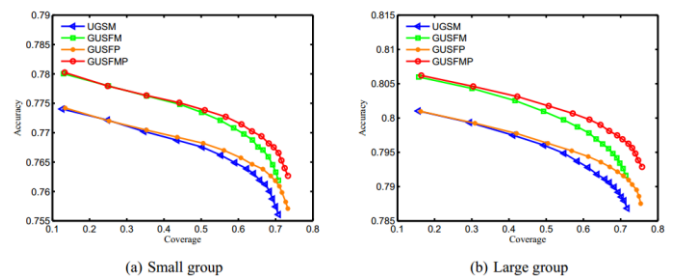


Fig. 2: Accuracy/Coverage comparison

4.4 PrevalenceAVG comparison

Figure 3 shows the results of the system PrevalenceAVG comparison for the small population (a) and the large population (b). Comparing the two graphs, the minimum of the PrevalenceAVG of the large groups was 108.15, while the minimum of the PrevalenceAVG of the small groups was 87.72. This indicates that the large group has better novelties regarding recommendations. At the same time, it can be found that in small or large group recommendations, the GUSFMP algorithm proposed in this paper is equivalent to the GUSFP algorithm, but it is significantly higher than that of the UGSM and GUSFM algorithms.

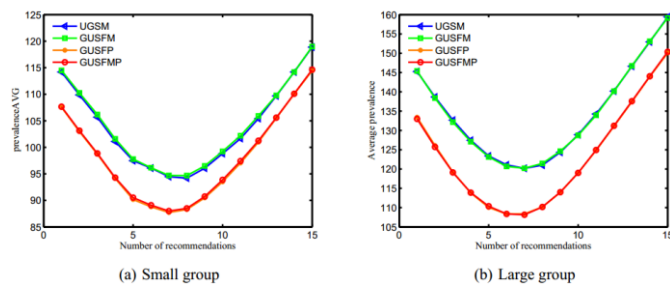


Fig. 3: PrevalenceAVG comparison

Conclusion

With the development of the Internet, recommendation systems in the individual user's personalized recommendation service has achieved great success. However, in some specific scenarios it is desirable to provide referral services to a group. This paper presents a collaborative filtering algorithm for mining group's interest. The group partitioning method is analyzed, the role of the group members in the group, the sparsity of the data and the popularity of the project are all considered, and the multiple indexes of the recommendation quality are analyzed. The comparison experiment proves the effectiveness of the algorithm GUSFMP. The algorithm can provide more a satisfactory recommendation service for the group users. Whether it is a spontaneously formed group or a systematically divided population, the members should have similar interests and hobbies. Future work will further study group composition to achieve more accurate grouping and to achieve more satisfactory results.

Acknowledgements

This work is supported by the Natural Science Foundation of Southwest University(NO.20120437). The authors also gratefully acknowledge the helpful comments and suggestions of reviewers, which have improved the presentation.

References

- [1] Tong Q L, Park Y, Park Y T, An empirical study on effectiveness of temporal information as implicit ratings[J], Expert Systems with Applications, vol.36, no.2, pp.1315-1321, (2009)
- [2] Hwang R H, Hsueh Y L, Chen Y T, An effective taxi recommender system based on a spatio-temporal factor analysis model[J], Information Sciences, vol.314, no.C, pp.429-433, (2014)
- [3] Lin C, Xie R, Guan X, et al, Personalized news recommendation via implicit social experts[J], Information Sciences, vol.254, no.1, pp.1-18, (2012)
- [4] Oh J, Kim S, Kim J, et al, When to recommend: A new issue on TV show recommendation[J], Information Sciences, vol.280, pp.261-274, (2014)
- [5] Mccarthy J F, Anagnost T D, MusicFX: an arbiter of group preferences for computer supported collaborative workouts[C], ACM(ACM Conference on Computer Supported Cooperative Work), pp.363-372, (2000)
- [6] O'Connor M, Dan C, Konstan J A, et al, PolyLens: A Recommender System for Groups of Users[M], Springer Netherlands(ECSCW 2001), pp.199-218, (2001)
- [7] Chao D L, Balthrop J, Forrest S. Adaptive Radio: Achieving Consensus Using Negative Preferences[C], Sanibel Island, Florida, United States(Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work), pp.120-123, (2004)
- [8] Ortega F, Bobadilla J, Hernando A, et al, Incorporating group recommendations to recommender systems: Alternatives and performance[J]. Information Processing & Management, vol.49, no.4, pp.895-901, (2013)
- [9] Ghazarian S, Nematbakhsh M A, Enhancing memory-based collaborative filtering for group recommender systems[J]. Expert Systems with Applications, vol.42, no.7, pp.3801-3812, (2015)
- [10] Ortega F, Hernando A, Bobadilla J, et al, Recommending items to group of users using Matrix Factorization based Collaborative Filtering[J]. Information Sciences, vol.345, no.C, pp.313-324, (2016)
- [11] Adomavicius G, Tuzhilin A, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions[J]. IEEE Transactions on Knowledge & Data Engineering, vol.17, no.6, pp.734-749, (2005)
- [12] Leng Y, Liang C, Zhang E, et al, A Collaborative Filtering Recommendation Algorithm Based on Item-Class Preference[J], Journal of the China Society for Scientific & Technical Information, (2011)
- [13] Mcqueen B J, Some methods for classification and analysis of multivariate observations[C], Statistics and Probability(In Proceedings of 5 th Berkeley Symposium on Mathematics), (2010)