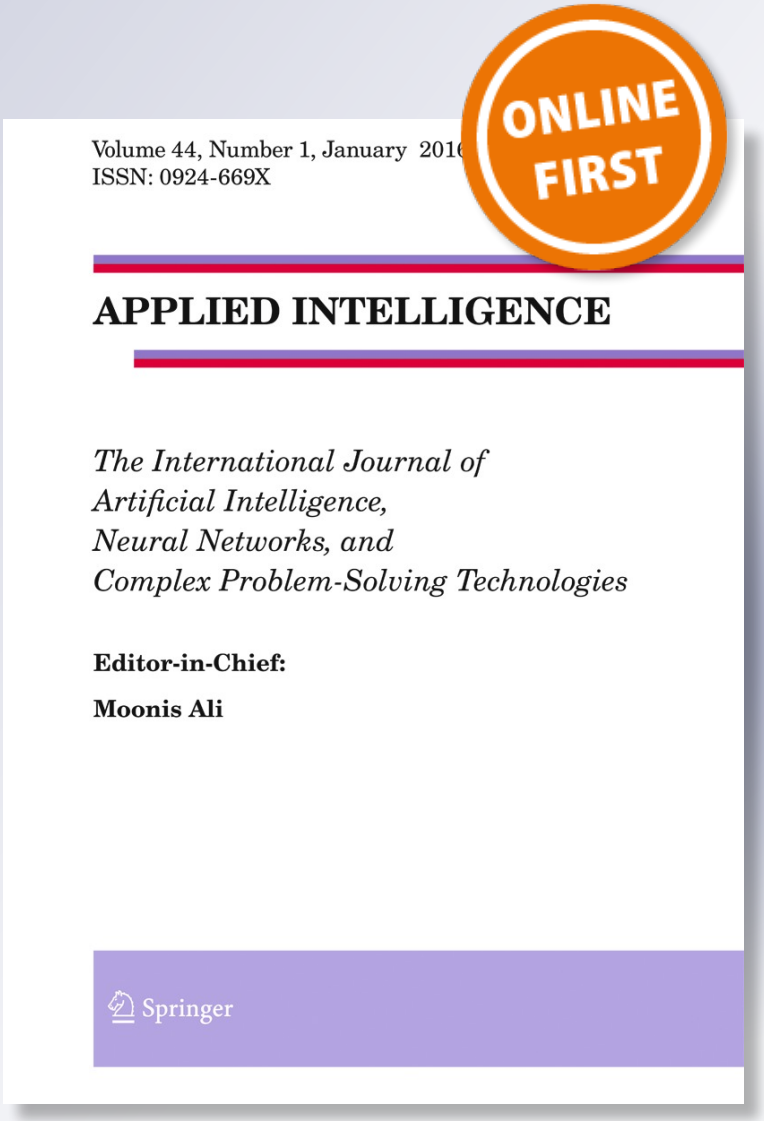
An effective collaborative filtering algorithm based on user preference clustering

Jia Zhang, Yaojin Lin, Menglei Lin & Jinghua Liu

Applied Intelligence
The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

ISSN 0924-669X

Appl Intell
DOI 10.1007/s10489-015-0756-9



Springer

Appl Intell
DOI 10.1007/s10489-015-0756-9

An effective collaborative filtering algorithm based on user preference clustering

Jia Zhang[1] · Yaojin Lin[1] · Menglei Lin[1] · Jinghua Liu[1]

Abstract Collaborative filtering is one of widely used recommendation approaches to make recommendation services for users. The core of this approach is to improve capability for finding accurate and reliable neighbors of active users. However, collected data is extremely sparse in the user-item rating matrix, meanwhile many existing similarity measure methods using in collaborative filtering are not much effective, which result in the poor performance. In this paper, a novel effective collaborative filtering algorithm based on user preference clustering is proposed to reduce the impact of the data sparsity. First, user groups are introduced to distinguish users with different preferences. Then, considering the preference of the active user, we obtain the nearest neighbor set from corresponding user group/user groups. Besides, a new similarity measure method is proposed to preferably calculate the similarity between users, which considers user preference in the local and global perspectives, respectively. Finally, experimental results on two benchmark data sets show that the proposed algorithm is effective to improve the performance of recommender systems.

Keywords Recommender systems · Collaborative filtering · User preference · Similarity · Clustering

✉ Yaojin Lin
yjlin@mnnu.edu.cn

Jia Zhang
zhangjia_gl@163.com

Menglei Lin
menglei36@126.com

Jinghua Liu
zzliujinghua@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, People's Republic of China

1 Introduction

With the development of the internet technologies, a deluge of data from all walks of life results in information overload problem [1, 12, 23, 31]. To address this problem, many large web sites and e-commerce sites exploit various convenient and efficient recommender systems to improve service quality with the aim to attract and retain loyal users. Such as the recommendation of books in Amazon [10], applications in markets [13], videos in YouTube [4], and results in the web search [41].

Collaborative filtering is one of successful techniques in recommender systems, which is to recommend items for a user through analyzing the user's data, and the data can be obtained by tracking browsing history, purchasing records, and rating records, etc [7, 14, 16, 18, 24, 28]. With years of development, this recommendation technology can be mainly classified into two categories: the model-based approach and the memory-based approach [38]. The model-based approach first constructs a prediction model based on the user-item rating matrix, and then predicts ratings on target items. Differing from the former, the memory-based approach first calculates the similarity between users/items, and selects the top-k similar users/items as the neighbors of the active user/target item, and then generates the predicted results. In addition to collaborative filtering, the

Published online: 15 February 2016

content-based approaches [27, 30, 36], hybrid filtering [11, 33, 39], and demographic filtering [22, 32, 40] are also proposed with different applications. Furthermore, referred to the memory-based approach, which can be categorized into user-based or item-based. In this paper, we focus on improving the performance of recommender systems based on the user-based method to reduce the impact of the data sparsity [29, 34].

In previous related works, modifications and enhancements of collaborative filtering are mainly embodied in two aspects: the similarity measure modification and the neighbor selection[15, 20, 26, 42]. Aimed at the similarity measure modification, traditional similarity measure methods, such as Pearson Correlation Coefficient (PCC) [9], and Cosine (COS) [35] were widely used in recommender systems. Besides, Jamali and Ester [17] proposed a modified similarity measure method based on PCC using a sigmoid function (SPCC), which emphasized the importance of common rated items. Intuitively, if more common rated items exist between users, then they are more similar. According to the Cosine similarity measure method does not take the rating scale into account, and the adjusted Cosine similarity measure method (ACOS) [37] is proposed to solve the shortage. For example, Ahn [2] introduced a new heuristic similarity measure method, which considered three factors of the similarity measure: proximity, impact, and popularity of ratings, and thus, was called the PIP method. However, PIP is limited in considering the local rating information, and ignores the global user preference. Liu et al. [25] analyzed the shortage of PIP, and proposed a new heuristic similarity model (NHSM). NHSM not only inherits the advantage of the PIP method, but also pays attention to the proportion of common rated items and user preference. In addition to the above proposed similarity measure methods, researchers also have proposed many modified neighbor selection approaches. For example, Kaleli [19] proposed an entropy-based optimization of forming a more qualified neighbor set, which assigned a degree of uncertainty (DU) for every user, and demanded neighbors with minimum differences of the value of DU and maximum of the similarity with the active user. Boumaza and Brun [8] introduced a conception about global neighbors, which are neighbors of all active users. Kim and Yang [21] presented a threshold-based neighbor selection approach. In this approach, neighbors were determined in a certain selection range with respect to the similarity of the preferences. Anand and Bharadwaj [3] introduced a recommendation framework combining both local and global similarities to solve the data sparsity, which allows the variation of the importance given to the global user similarity with regards to the local user similarity.

In this paper, we present an effective collaborative filtering algorithm based on user preference clustering, which differs from aforementioned ones. On the one hand, user groups are introduced to select more accurate and reliable neighbors for the active user. As we know, users with different preferences have different rating habits. Therefore, users can be clustered into different user groups. (1) optimistic user group, in which users prefer to rate high marks; (2) pessimistic user group, in which users prefer to rate low marks; (3) neutral user group, in which users have the tendency to give reasonable marks for items. On the other hand, we notice that most of the previous similarity measure methods are not suitable for capturing user preference, and we propose a new similarity measure method to calculate the similarity between users in the process of clustering. Moreover, extensive experiments show that our proposed algorithm can significantly improve the performance with the sparse rating data. Finally, major contributions of this work can be summarized as follows:

- Users are allocated into different user groups based on user preference clustering.
- A new similarity measure method with the factor of user preference is proposed.
- Extensive experimental results show that our proposed method is effective.
- The proposed algorithm based on user preference clustering can be combined with other similarity measure methods freely.

The rest of this paper is organized as follows. We review traditional similarity measure methods and the user-based collaborative filtering approach in Section 2. And then, in Section 3, we deeply describe our proposed algorithm. Section 4 demonstrates and explains the experimental results. Finally, we conclude the work and give future work in Section 5.

2 Preliminaries

In recommender systems, a user-item rating matrix R is constructed to make recommendation for the active user, in which there are ratings of m users on n items, and U denotes the set of m users, I represents the set of n items. Note that rating data of the rating matrix is sparse, the missing or unknown rating data is denoted by the symbol ?, and r_{ui} denotes the rating of user u on item i.

According to the user information stored in rating matrix R, traditional similarity measure methods, such as PCC and COS, are widely used to calculate the similarity between

users in the user-based collaborative filtering approach, as given in (1) and (2) respectively:

$$sim(a,b)^{PCC} = \frac{\sum_{i\in I_{ab}}(r_{ai}-\overline{r}_a)\times(r_{bi}-\overline{r}_b)}{\sqrt{\sum_{i\in I_{ab}}(r_{ai}-\overline{r}_a)^2}\times\sqrt{\sum_{i\in I_{ab}}(r_{bi}-\overline{r}_b)^2}}, \tag{1}$$

$$sim(a,b)^{COS} = \frac{\sum_{i\in I_{ab}} r_{ai}\times r_{bi}}{\sqrt{\sum_{i\in I_{ab}} r_{ai}^2}\times\sqrt{\sum_{i\in I_{ab}} r_{bi}^2}}, \tag{2}$$

Where $sim(a,b)$ denotes the similarity between user a and user b, I_{ab} is the set of common rated items by user a and user b, r_{ai} is the rating of user a on item i, and $\overline{r}_a$ is the average rating of user a. After the similarity is calculated, the k nearest similar users are specified as the neighbors of the active user, then the prediction can be worked out on the target item. The recommended formula is defined as follow:

$$p_{ti} = \overline{r}_t + \frac{\sum_{u\in U_{nei}} sim(t,u)\times(r_{ui}-\overline{r}_u)}{\sum_{u\in U_{nei}}|sim(t,u)|}, \tag{3}$$

Where p_{ti} denotes the prediction of active user t on target item i, U_{nei} is the neighbor set of active user t, $|U_{nei}| = k$.

3 The proposed algorithm

In collaborative filtering, the traditional way of searching neighbors for the active user depends on the rating information of common rated items by two users. However, some shortages exist in the traditional collaborative filtering approach, i.e., the factor of user preference is not taken into account, and a small portion of collected users' data is utilized. In order to overcome these drawbacks, a novel effective collaborative filtering algorithm based on user preference clustering is proposed. The proposed algorithm's flowchart is shown in Fig. 1.

3.1 Clustering based on user preference

In practical recommender applications, users could have starkly different views on an item. For example, some users are kind and they might rate their likeable and favorite items with high marks. Conversely, some users have strict attitudes on the rating, who might tend to rate low marks. Finally, some users might give reasonable marks for different items. As discussed above, users could be allocated into three different user groups. Suppose C_o, C_p, and C_n represent optimistic user group, pessimistic user group, and neutral user groups respectively. Meanwhile, c_o is the clustering center of C_o, c_p is the clustering center of C_p, and c_n is the clustering center of C_n. Then, we introduce the selection of clustering centers.

Definition 1 (Different user preferences) Suppose U_h and U_l are two subsets of user set U. In which $U_h = \{u \in U|\overline{r}_u \geq \alpha\}$, and $U_l = \{u \in U|\overline{r}_u < \beta\}$. $c_o \in U_h$, and $c_p \in U_l$.

Where α is set as a high mark, β is set as a low mark. For example, in a 1–5 scale rating matrix, α can be set as 4, and β can be set as 2. Therefore, U_h is a subset of users who prefer to rate high marks on items. Similarly, users in the U_l tend to rate items with low marks.

Definition 2 (Maximum rating number) c_o is a user from the subset U_h who has maximum rating number; c_p is a user from the subset U_l who has maximum rating number.

The above Definitions give two criteria for the selection of clustering centers, i.e., the expected c_o should be with the preference to rate high marks, meanwhile, c_o should have as many ratings as possible on items. Through these criteria, we can judge a user's preference via calculating the similarity between the user and all cluster centers. These cluster centers of different user groups are defined as follow:

Definition 3 Clustering center c_o of optimistic user group can be uniquely determined, as follow:

$$c_o = u \Leftarrow \arg\max_u |I_u|, \tag{4}$$

where $\forall u \in U_h$, $I_u = \{i \in I|r_{ui} \neq ?\}$. Clustering center c_p of pessimistic user group can be uniquely determined, as follow:

$$c_p = u \Leftarrow \arg\max_u |I_u|, \tag{5}$$

where $\forall u \in U_l$, $I_u = \{i \in I|r_{ui} \neq ?\}$.

If $\forall i \in I$, then the rating of c_n on i is $\overline{r}_i$, $\overline{r}_i$ is the average rating of item i. With this, clustering center c_n of neutral user group is constructed.

From Definition 3, we know c_o, c_p, and c_n are uniquely determined and beneficial for achieving user preference clustering, in which both c_o and c_p are from user set U, and the difference between c_o and c_p is that they are with totally opposite preference. c_n is virtual and constructed for obtaining the neutral user group. In consideration of an enormous amount of users stored in the user-item rating matrix, the average rating on an item can represent the majority view on this item. Therefore, c_n can be regarded as a typical user who prefers to give reasonable marks. Based

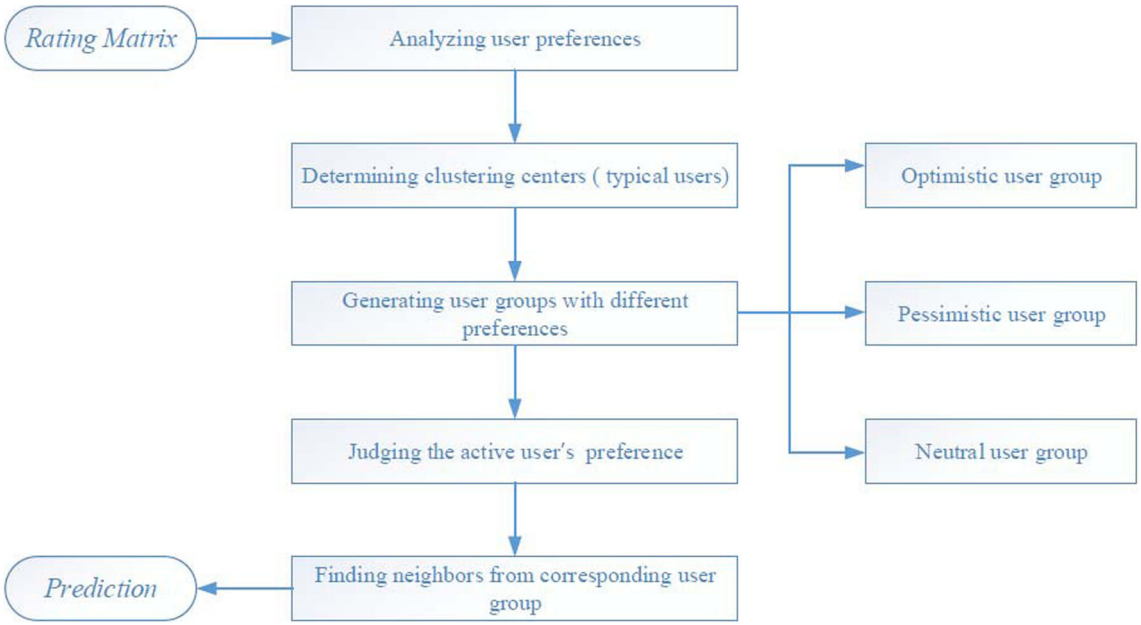


Fig. 1 Algorithm's flowchart

on this, three typical users are found as three cluster centers respectively, who have different characteristics for generating user groups. Then, we can obtain user groups with different preferences in Definition 4.

Definition 4 Suppose $C = \{c_o, c_p\}$, $\forall u \in U - C$, the preference of u is determined as follow:

$u \in C_o$, if u satisfies $sim(u, c_o) > sim(u, c_p)$, and $sim(u, c_o) > sim(u, c_n)$;
$u \in C_p$, if u satisfies $sim(u, c_p) > sim(u, c_o)$, and $sim(u, c_p) > sim(u, c_n)$;
$u \in C_n$, if u satisfies $sim(u, c_n) > sim(u, c_o)$, and $sim(u, c_n) > sim(u, c_p)$.

From Definition 4, we can easily identify different preferences of users based on the similarity between users, and users with the consistent preference are assigned to the same user group. Therefore, different user groups can be obtained, i.e., optimistic user group U_o, pessimistic user group U_p, and neutral user group U_n.

3.2 User similarity

In the process of clustering, the rating information of clustering centers is with special characteristics, i.e., c_o prefers to rate high marks, and the determination of a user's preference depends on the similarity between the user and these clustering centers. Therefore, an effective similarity measure method is helpful for assigning the remaining users into different user groups. In order to highlight the importance of user preference, we propose a new similarity measure method to calculate the similarity between users, as follow:

$$sim(a, b)^{UPS} = \exp\left(-\frac{\sum_{i \in I_{ab}} |r_{ai} - r_{bi}|}{|I_{ab}|} \times |\bar{r}_a - \bar{r}_b|\right) \times \frac{|I_a| \cap |I_b|}{|I_a| \cup |I_b|}, \quad (6)$$

From (6), we know that two important factors are involved. In the global perspective, user preference is reflected by calculating the average rating on all items, and the higher the difference of average ratings between users, the more different preferences of them are shown. Locally, the factor of common rated items are taken into account to reflect the difference between user preferences. Users who have more common rated items with less difference between their preferences, the higher their similarity is shown. Therefore, users who have consistent preferences are easily assigned to the same user group.

Table 1 shows an example of a user-item rating matrix, in which u_1-u_5 are users and i_1-i_9 are items. We can calculate

Table 1 An example of a user-item rating matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
u_1	1	2	?	3	2	?	2	?	?
u_2	2	4	4	?	4	?	?	2	3
u_3	5	5	?	4	?	4	3	?	4
u_4	5	?	5	4	4	?	4	4	?
u_5	1	?	?	?	2	?	?	?	2

the similarity between users in Table 1 by different similarity measure methods mentioned in Section 1, as shown in Fig. 2. In Fig. 2, since the user similarity matrix is symmetric, and partial similarity values are not demonstrated.

Figure 2a gives the user similarity matrix according to the COS method. From Table 1, we can see that u_1 and u_5 have similar ratings, both of them prefer to rate low marks, and thus most of the ratings in u_2 are 4. However, the similarity between u_1 and u_2 is 1 in Fig. 2a. This drawback also exists in the SPCC method, as shown in Fig. 2b. For example, Fig. 2b shows that u_1 and u_2 can obtain the highest correlation regardless of user preference. Compared with COS and SPCC, the computational similarities are more accurate by the NHSM method, as shown in Fig. 2c. But from Fig. 2c, we can see that the similarity between u_2 and u_4 is higher than the similarity between u_3 and u_4, however, u_3 and u_4 have more similar preference in fact. Figure 2d gives the user similarity matrix according to the proposed UPS method. In Fig. 2d, we notice that the similarity between u_1 and u_5 is high. In addition, both u_3 and u_4 prefer to rate high marks, and their similarity is 0.3153, which is higher than the similarity between u_2 and u_4. Based on these observations, we can conclude that the proposed similarity measure method is more suitable for depicting the characteristic of user preference.

3.3 Recommendation method

In this section, we design the related algorithm to make recommendations for the active user. From the analysis in Sections 3.1–3.2, we first calculate the similarity between users by our proposed method, and the similarity matrix is denoted by sim^{UPS}. Then, c_o, c_p, and c_n are determined as clustering centers with different preferences respectively. Finally, users are assigned to different user groups based on the user similarity. With this, different user groups are generated, which are optimistic user group U_o, pessimistic user group U_p, and neutral user group U_n. After the process of clustering is finished, we can obtain k nearest neighbors for the active user, and the neighbor selection approach is defined as follows:

Definition 5 Suppose U_{nei} is the neighbor set of active user t, $|U_{nei}| = k$, and $U_{nei} \subset U$. If $t \in U_o$, then $U_{nei} \subset U_o \cup U_n$. If $t \in U_p$, then $U_{nei} \subset U_p \cup U_n$. If $t \in U_n$, then $U_{nei} \subset U_n$.

From Definition 5, we know that a user from U_n could possibly become a neighbor of all active users, this is because users from U_n have reasonable ratings and are valuable for predicting unrated items. Inversely, users from U_o (or U_p) who prefer to rate high marks (or low marks), all of them can't become neighbors of the active user who is from U_n. After neighbor set U_{nei} is obtained for active user t, we can predict the rating p_{ti}, as follow:

$$p_{ti} = \bar{r}_t + \frac{\sum_{u \in U_{nei}} sim^{UPS}(t,u) \times (r_{ui} - \bar{r}_u)}{\sum_{u \in U_{nei}} |sim^{UPS}(t,u)|}, \tag{7}$$

In order to provide a clear description, we display our proposed method in Algorithm 1.

Algorithm 1 Collaborative filtering algorithm based on user preference clustering(UPUC-CF)

Input: the rating matrix R, threshold values: α and β.
Output: the prediction p_{ti} of active user t.

1: use Eq. (6) to calculate the similarity between users, and the similarity matrix generated is denoted by sim^{UPS}.
2: determine c_o, c_p, and c_n as clustering centers according to Definition 3.
3: generate optimistic user group U_o, pessimistic user group U_p, and neutral user group U_n according to Definition 4.
4: obtain the neighbor selection range of active user t according to Definition 5, and then generate neighbor set U_{nei} based on the similarity between users.
5: use Eq. (7) to predict rating p_{ti}.

To evaluate the performance of our proposed algorithm, the analysis of time complexity is imperative. The selection of clustering centers requires the extra time cost is $O(m)$, where m denotes the number of users, and when we calculate the similarity between users by the proposed method, the computational complexity is $O(m(m+2))$. As a whole, although the time complexity of the similarity calculation has increased slightly compared with traditional user-based recommendation algorithm (i.e., $O(m^2)$), it is usual that the similarity is computed off-line to lessen the time complexity burden.

4 Experiments

4.1 Data sets

We test our proposed algorithm on two well-known data sets, MovieLens (ML) and HetRec2011-MovieLens (HRML). The ML data set was collected by GroupLens research team at the University of Minnesota, in which 943 users rated on 1682 movies with 100000 ratings, and each user at least had 20 rating recode on movies. The density of ML data set is 6.3047 %. The second data set HRML was released on the 2nd international workshop on information

heterogeneity and fusion in recommender systems. We randomly drew 1036 users and 1300 movies from HRML data set, the total number of ratings is 106210. And the sparsity of extracted data set is 92.1139 %. In addition, each data set is divided into 5 groups. Twenty percent of all data is selected as the test set, and the remaining data is the training set. For the impartial of experimental results, we adopt the 5-fold cross-validation by choosing different test set and training set.

4.2 Evaluation metrics

To date, researchers have presented many metrics to evaluate the performance of recommender systems [5, 6]. Generally, evaluation metrics are classified into two categories: (1) evaluation metrics of the prediction quality, such as mean absolute error (MAE), coverage, and accuracy; (2) evaluation metrics of the recommendation quality, such as precision, recall, and novelty. In order to estimate the performance of our proposed method, we utilize the MAE and coverage to measure the prediction quality, and the precision and recall to measure the quality of the recommendation set.

MAE The MAE is one of the most widely used metrics to evaluate the recommendation accuracy, and is defined as the average of absolute difference between prediction values and actual ratings. The lower the MAE reflects, the more accurate predictions. Assuming $I_u = \{i \in I | p_{ui} \neq ? \wedge r_{ui} \neq ?\}$, which is the set of items rated by user u having prediction values, p_{ui} is the prediction of user u on item i. The MAE is calculated as follows:

$$MAE = \frac{1}{\#U} \sum_{u \in U} \frac{\sum_{i \in I_u} |p_{ui} - r_{ui}|}{\#I_u}, \tag{8}$$

Coverage The coverage indicates the proportion of predicted items from the total number of items, which applies to the recommender system to reflect the capacity of the prediction. This metric should be as high as possible for good prediction quality. Assuming there are n items, the number of predicted items is s, the coverage is calculated as follows:

$$Coverage = \frac{s}{n}, \tag{9}$$

Precision The precision is the proportion of relevant recommendations from the total number of recommendations. The higher the precision denotes, the better the recommendation performance. Assuming Z_u is the set of top-N recommendations to user u, θ is set as a relevancy threshold, and θ equals the median value of ratings from the rating matrix. The precision is calculated as follows:

$$Precision = \frac{1}{\#U} \sum_{u \in U} \frac{\#\{i \in Z_u | r_{ui} \geq \theta \wedge p_{ui} \geq \theta\}}{N}, \tag{10}$$

Recall The recall is the average proportion of relevant recommendations from the total number of relevant items that user actually liked according to actual ratings. This metric is also as high as possible for good recommendation performance. Assuming T_u is the number of relevant items in the test set, which are liked by user u. The recall is calculated as follows:

$$Recall = \frac{1}{\#U} \sum_{u \in U} \frac{\#\{i \in Z_u | r_{ui} \geq \theta \wedge p_{ui} \geq \theta\}}{\#T_u}. \tag{11}$$

4.3 Experimental results

In order to verify the effectiveness of the proposed algorithm, we first explain that the fact of user preference is

Fig. 2 Similarity matrixs of users in Table 1

(a) COS

	u_1	u_2	u_3	u_4	u_5
u_1	1.0000	1.0000	0.8981	0.9104	1.0000
u_2		1.0000	0.9600	0.9429	0.9904
u_3			1.0000	0.9928	0.9080
u_4				1.0000	0.9080
u_5					1.0000

(b) SPCC

	u_1	u_2	u_3	u_4	u_5
u_1	1.0000	0.8176	0.3756	−0.7192	0.7311
u_2		1.0000	0	0	0.7080
u_3			1.0000	0.7080	−0.7311
u_4				1.0000	−0.7311
u_5					1.0000

(c) NHSM

	u_1	u_2	u_3	u_4	u_5
u_1	1.0000	0.0044	0.0062	0.0055	0.0082
u_2		1.0000	0.0041	0.0086	0.0051
u_3			1.0000	0.0066	0.0022
u_4				1.0000	0.0030
u_5					1.0000

(d) UPS

	u_1	u_2	u_3	u_4	u_5
u_1	1.0000	0.0537	0.0044	0.0030	0.3333
u_2		1.0000	0.0630	0.0869	0.0677
u_3			1.0000	0.3153	0.0002
u_4				1.0000	0.0001
u_5					1.0000

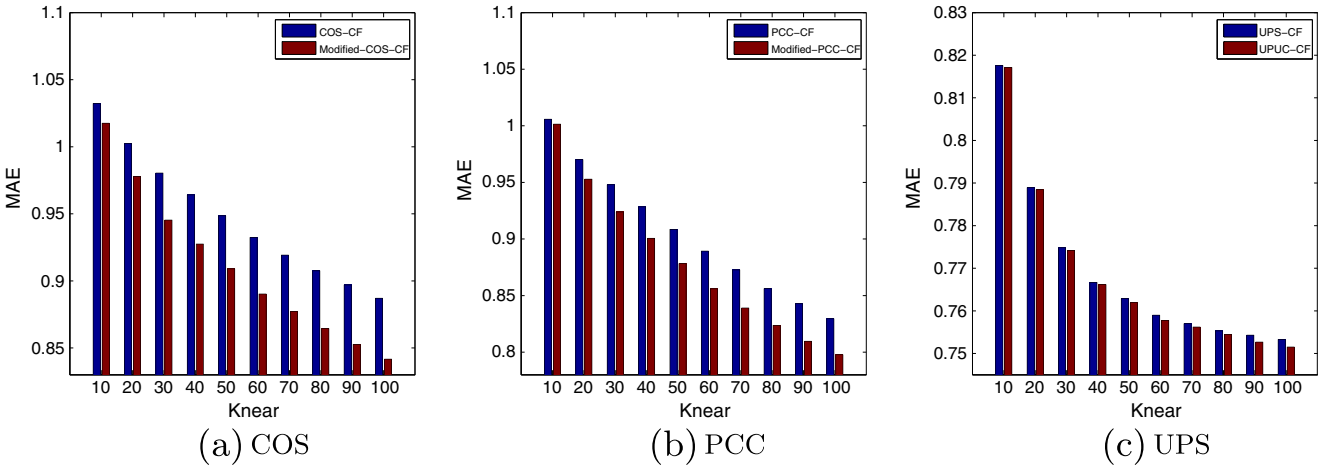


Fig. 3 Validity check of user preference clustering on ML data set

conducive to the performance improvement, and several experiments are performed on two benchmark data sets, as shown in Figs. 3–4. Since the number of the nearest neighbor can affect the performance of recommendation algorithms, therefore the number of the nearest neighbor k varies from 10 to 100 in experiments.

Figure 3 demonstrates experimental results on ML data set. In Fig. 3a, the result labeled as COS-CF shows the MAE of traditional COS-based collaborative filtering algorithm [35], and Modified-COS-CF presents the MAE of modified COS-CF with user preference, which first generates three user groups by our proposed method, and then forms the nearest neighbor set from corresponding user group/user groups based on the similarity calculated by COS. According to the result comparison of COS-CF and Modified-COS-CF, we can see that the recommendation accuracy of COS-CF is lower than that of Modified-COS-CF with the increasing of the number of the nearest neighbors. Similarly, Modified-PCC-CF is also clearly superior to traditional PCC-based collaborative filtering (PCC-CF) [9], as shown in Fig. 3b. Therefore, we can conclude from Fig. 3a, b that the performance of Modified-COS-CF and Modified-PCC-CF have significant improvement compared with COS-CF and PCC-CF. Figure 3c demonstrates the result comparison of a recommendation algorithm based on the UPS method (UPS-CF) and its modified approach (UPUC-CF). From Fig. 3c, we can see the MAE of UPUC-CF is lower in the whole top-k range, however, since user preference clustering of UPUC-CF depends on the similarity estimated by the UPS method, the difference of the results is delicate. In addition, Fig. 4 demonstrates experimental results on HRML data set. From Fig. 4, we can get the same

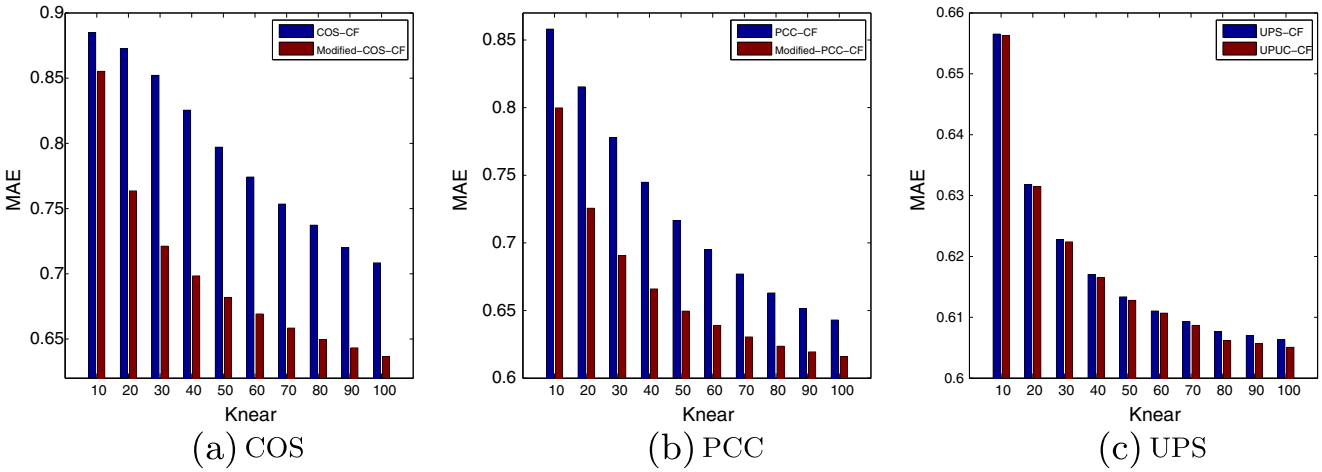


Fig. 4 Validity check of user preference clustering on HRML data set

results as Fig. 3: all of modified approaches have higher recommendation accuracy than traditional algorithms based on single similarity computation. Therefore, we can draw the conclusion that the recommendation algorithm based on user preference is more effective.

For further validating our proposed algorithm, we compare our proposed algorithm with some state-of-the-art recommendation algorithms, i.e., COS-CF [35], PCC-CF [9],SPCC-CF [17], and NHSM-CF [25]. Results of the different algorithm comparisons are shown in Figs. 5–6.

Figure 5 shows the performance of different algorithms with the number of the nearest neighbors on ML data set. In which Fig. 5a, b, c, d demonstrate the results of MAE, coverage, precision, and recall respectively. In Fig. 5a, the MAE of all algorithms decrease with the increasing of the number of the nearest neighbors. We can conclude that our proposed algorithm obtains the better MAE in the whole top-k range. In Fig. 5b, our proposed algorithm has remarkable improvement by comparing other algorithms, and the coverage is more than 85 % when the number of the nearest neighbors is 10. In short, our proposed algorithm has better prediction quality on ML data set. In addition, Fig. 5c, d show that the number of recommended items equally increase with the increasing of the number of the nearest neighbors, i.e., the number of the recommended item also varies from 20 to 30 when the k is from 20 to 30. In Fig. 5c, d, both the precision and recall of the proposed algorithm always have the better results with the increasing of the number of the nearest neighbors. Therefore, the effectiveness of the proposed algorithm can be verified to improve the recommendation performance on ML data set.

Figure 6 demonstrates experimental results on HRML data set. From Fig. 6, we can conclude that the change situation of different algorithms with the increasing of the number of the nearest neighbors is basically the same as Fig. 5. Differing from Fig. 5, Fig. 6 shows NHSM-CF and SPCC-CF compare unfavorably with PCC-CF in terms of the prediction quality and recommendation performance, when the

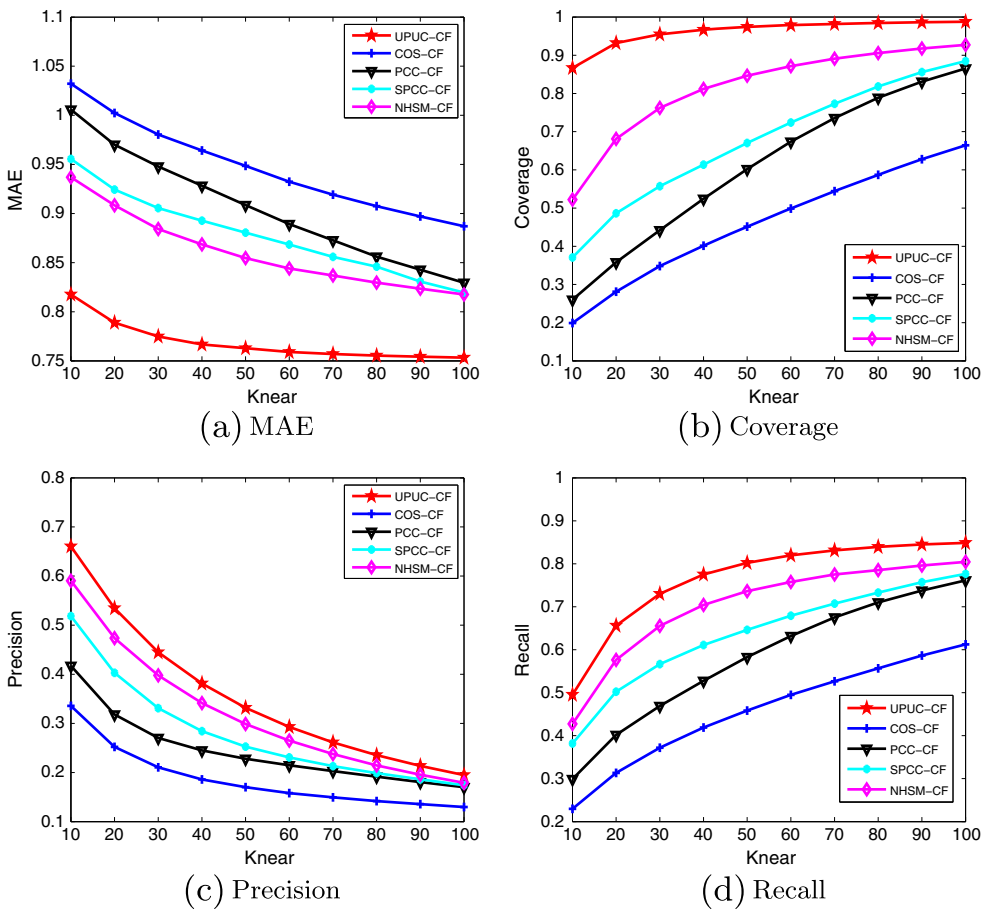


Fig. 5 Algorithms comparison based on ML data set

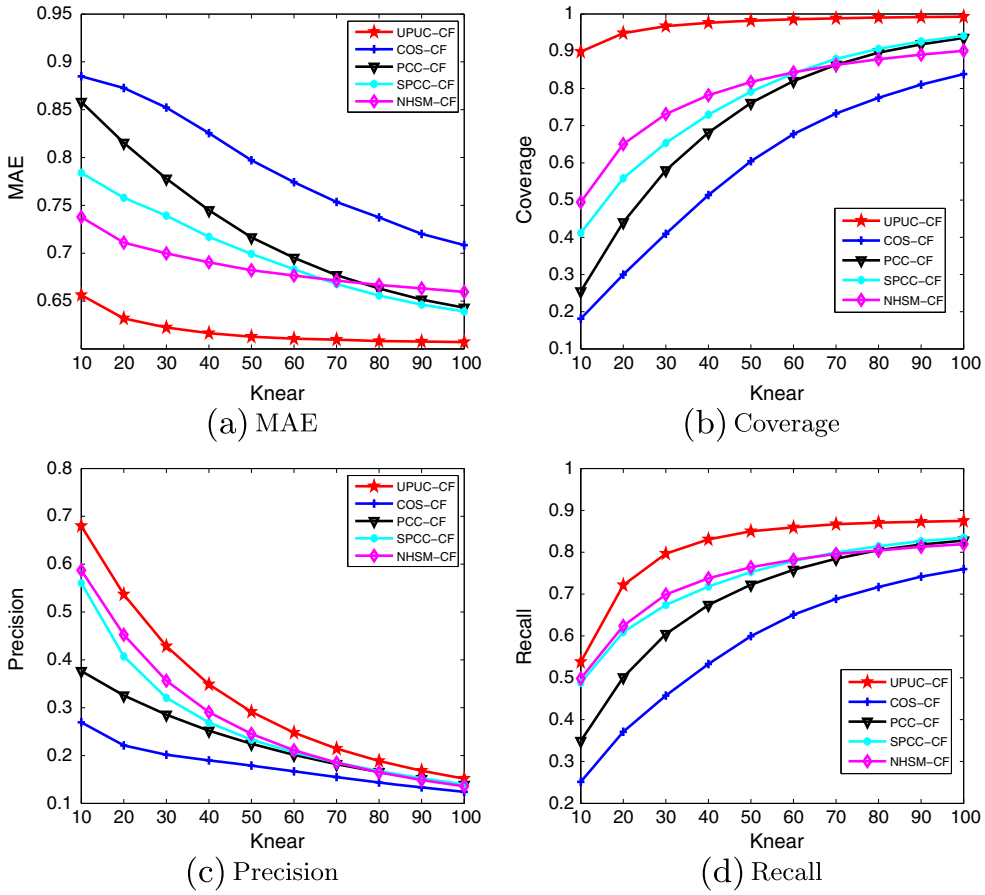


Fig. 6 Algorithms comparison based on HRML data set

number of the nearest neighbors is more than 70. Therefore, experiments on HRML data set reveal that our proposed algorithm is superior to other algorithms.

From Figs. 5–6, we can conclude that our proposed algorithm can obtain the better prediction quality and recommendation performance than some other methods. As previous methods mentioned in Section 1, many modifications and enhancements of collaborative filtering are published aiming to discover more accurate and reliable neighbors for improving the performance of recommender systems. Although the purpose of our work is similar with them, we propose a novel collaborative filtering algorithm based on user preference clustering. This method considers users who have different rating habits, and different typical users are defined to generate user groups with different preferences. Also, a new similarity measure method is proposed, which not only considers the rating information on common rated items by users, but also is up to the global information of user preference. In short, experimental results on two benchmark data sets show that the proposed algorithm is effective to improve the prediction quality and recommendation performance.

5 Conclusion and future work

In this paper, we introduced a collaborative filtering algorithm based on user preference clustering. Our approach is based on an assumption that users have different rating habits. For distinguishing different typical users, the primary work in this paper is to design a framework to assign users into user groups with different preferences. Therefore, the neighbor users of the active user can be found with consistent preference. As we know, the traditional Pearson Correlation Coefficient and Cosine similarity measure methods have a shortage in that they only consider the factor of commonly rated items by users. To solve this problem, we proposed a new similarity measure method to consider user preference from the local and global perspectives respectively. In addition, an example was illustrated in our paper,

which has proved that the proposed similarity measure method is more effective and suitable for calculating the similarity between users. In experiments, we evaluated the effectiveness of our proposed algorithm on quality and recommendation performance improvement respectively, and experimental results on two benchmark data sets demonstrated our proposed algorithm has better performance compared with some state-of-the-art recommendation algorithms. In a word, the proposed algorithm is effective to improve the performance for recommender systems.

In the future we will continue to analyze the impact of the user behavior in recommender systems, and study the mechanism of the user rating behavior.

Acknowledgments The authors would like to thank the anonymous reviewers and the editor for their constructive and valuable comments. This work is supported by grants from the National Natural Science Foundation of China (Nos. 61303131, and 61379021), the Department of Education of Fujian Province (No. JA14129), and the Program for New Century Excellent Talents in Fujian Province University.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
2. Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci 178(1):37–51
3. Anand D, Bharadwaj KK (2011) Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. Expert Syst Appl 38(5):5101–5109
4. Baluja S, Seth R, Sivakumar D, Jing Y, Yagnik J, Kumar S, Ravichandran D, Aly M (2008) Video suggestion and discovery for YouTube: taking random walks through the view graph. In: Proceedings of the International Conference on World Wide Web, pp 895–904
5. Bobadilla J, Ortega F, Hernando A, Gutirrez A (2013) Recommender systems survey. Knowl-Based Syst 46:109–132
6. Bobadilla J, Hernando A, Ortega F, Bernal J (2011) A framework for collaborative filtering recommender systems. Expert Syst Appl 38(12):14609–14623
7. Bobadilla J, Hernando A, Ortega F, Gutirrez A (2012) Collaborative filtering based on significances. Inf Sci 185:1–17
8. Boumaza AM, Brun A (2012) Stochastic search for global neighbors selection in collaborative filtering. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. ACM, New York, USA, pp 232–237
9. Breese JS, Heckerman D, Kadie CM (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence. Madison, USA, pp 43–52
10. Brynjolfsson E, Hu Y, Smith MD (2003) Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers. Manag Sci 49(11):1580–1596
11. Choi K, Yoo D, Kim G, Suh Y (2012) A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. Electr Commer Res Appl 11(4):309–317
12. Choi K, Suh Y (2013) A new similarity function for selecting neighbors for each target item in collaborative filtering. Knowl-Based Syst 37:146–153
13. Costa-Montenegro E, Barragns-Martnez AB, Rey-Lpez M (2012) Which App? A recommender system of applications in markets: Implementation of the service for monitoring users' interaction. Expert Syst Appl 39(10):9367–9375
14. Goldberg D, Nichols DA, Oki BM, Terry DB (1992) Using collaborative filtering to weave an information tapestry. Commun ACM 35(12):61–70
15. Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 230–237
16. Huete JF, Fernndez-Luna JM, de Campos LM, Rueda-Morales MA (2012) Using past-prediction accuracy in recommender systems. Inf Sci 199:78–92
17. Jamali M, Ester M (2009) TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 397–406
18. Jia C-X, Liu R-R (2015) Improve the algorithmic performance of collaborative filtering by using the interevent time distribution of human behaviors. Physica A 436:236–245
19. Kaleli C (2014) An entropy-based neighbor selection approach for collaborative filtering. Knowl-Based Syst 56:273–280
20. Kim H-N, Ji A-T, Ha I, Jo G (2010) Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electr Commer Res Appl 9(1):73–83
21. Kim T-H, Yang S-B (2007) An effective threshold-based neighbor selection in collaborative filtering. In: Proceedings of the 29th European conference on IR Research. ECIR'07. Springer, Berlin, Heidelberg, pp 712–715
22. Krulwich B (1997) Lifestyle finder: intelligent user profiling using large-scale demographic data. Artificial Intelligence Magazine 18(2):37–45
23. Li Y, Zhai C, Chen Y (2014) Exploiting rich user information for one-class collaborative filtering. Knowl Inf Syst 38(2):277–301
24. Koohborfardhaghighi S, Kim J (2013) Using structural information for distributed recommendation in a social network. Appl Intell 38(2):255–266
25. Liu H, Hu Z, Mian AU, Tian H, Zhu X (2014) A new user similarity model to improve the accuracy of collaborative filtering. Knowl-Based Syst 56:156–166
26. Liu Q, Chen E, Xiong H, Ding CHQ, Chen J (2012) Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Trans Syst Man Cybern B 42(1):218–233
27. Liu N-H (2013) Comparison of content-based music recommendation using different distance estimation methods. Appl Intell 38(2):160–174
28. Luo X, Xia Y, Zhu Q (2012) Incremental collaborative filtering recommender based on regularized matrix factorization. Knowl-Based Syst 27:271–280
29. Massa P, Avesani P (2004) Trust-aware collaborative filtering for recommender systems. Lect Notes Comput Sci 3290:492–508
30. Meteren R, Someren M (2000) Using content-based filtering for recommendation. In: Proceedings of ECML 2000 Workshop: Maching Learning in Information Age, pp 47–56
31. Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. Expert Syst Appl 39(11):10059–10072

32. Pazzani MJ (1999) A framework for collaborative, content-based, and demographic filtering. Artificial Intelligence Review-Special Issue on Data Mining on the Internet 13(5-6):393–408
33. Porcel C, Tejeda-Lorente A, Martnezb MA, Herrera-Viedma E (2012) A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. Inf Sci 184(1):1–19
34. Ramezani M, Moradi P, Akhlaghian F (2014) A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains. Physica A 408:72–84
35. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work. Chapel Hill, USA, pp 175–186
36. Salter J, Antonopoulus N (2006) CinemaScreen recommender agent: combining collaborative and content-based filtering. IEEE Intell Syst 21(1):35–41
37. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp 285–295
38. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput Surv 47(1):3:1–3:45
39. Shinde SK, Kulkarni UV (2012) Hybrid personalized recommender system using centering-bunching based clustering algorithm. Expert Syst Appl 39(1):1381–1387
40. Vozalis MG, Margaritis KG (2007) Using SVD and demographic data for the enhancement of generalized collaborative filtering. Inf Sci 177(15):3017–3037
41. Zhang X, Li Y (2008) Use of collaborative recommendations for web search: an exploratory user study. J Inf Sci 34(2):145–161
42. Zhu T, Ren Y, Zhou W, Rong J, Xiong P (2014) An effective privacy preserving algorithm for neighborhood-based collaborative filtering. Futur Gener Comput Syst 36:142–155

Jia Zhang is currently working toward the Master degree from the School of Computer Science, Minnan Normal University. His research interests are focused on data mining.

Yaojin Lin received the Ph.D. degree in School of Computer and Information from Hefei University of Technology. He currently is an associate professor with Minnan Normal University and a postdoctoral fellow with Tianjin University. His research interests include data mining, and granular computing. He has published more than 40 papers in many journals, such as Neurocomputing, Decision Support Systems, Information Sciences, and Applied Intelligence.

Menglei Lin received the MS degree in Mathematics from Xiamen University. He currently is a professor in the School of Computer Science, Minnan Normal University. His research interests include data mining, and granular computing.

Jinghua Liu is currently working toward the Master degree from the School of Computer Science, Minnan Normal University. Her research interests are focused on data mining and granular computing.