



# A Hybrid Collaborative Filtering Recommendation Algorithm Using Double Neighbor Selection

Wenan Tan<sup>1,2(✉)</sup>, Xiaofan Qin<sup>1</sup>, and Qing Wang<sup>1</sup>

<sup>1</sup> School of Computer Science and Technology,  
Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China  
{wtan, qinxfan}@foxmail.com

<sup>2</sup> School of Computer and Information, Shanghai Polytechnic University,  
Shanghai 201209, China

**Abstract.** The traditional collaborative filtering algorithms are more successful used for personalized recommendation. However, the traditional collaborative filtering algorithm usually has issues such as low recommendation accuracy and cold start. Aiming at addressing the above problems, a hybrid collaborative filtering algorithm using double neighbor selection is proposed. Firstly, according to the results of user's dynamic similarity calculation, the similar interest sets of the target users may be dynamically selected. Analyzing the dynamic similar interest set of the target user, we can divide the users into two categories, one is an active user, and the other is a non-active user. For the active user, by calculating the trust degree of the users with similar interests, we can select the user with the trust degree of TOP-N, and recommend the target user. For the non-active user, the neighbor user may be found according to the similarity of the user on some attributes, and them with high similarity will be recommend to the target user. The experimental results show that the algorithm not only improves the recommending accuracy of the recommendation system, but also effectively solves the problem of data sparseness and user cold start.

**Keywords:** Collaborative filtering · Similarity · Dynamic neighbor selection · Trust · Hybrid model · Cold start

## 1 Introduction

As data growing rapidly, how to recommend products or information services to customers has attracted widespread attention from researchers at home and abroad. In order to help users quickly and accurately find what they want in larger amounts of data, the recommendation algorithms have appeared. The collaborative filtering recommendation algorithm is one of the most widely used [1]. The main principle is to find similar user sets or similar item sets for the recommended target users or target items based on the user-item rating in datasets. However, there are some drawbacks in the traditional collaborative filtering method. Firstly, the recommendation efficiency is declined when the data is extremely sparse or new users and new items appear due to excessive dependence on the user's rating of the item. Secondly, the recommendation

is assumed that the users are independent and equally distributed, and it ignores the credibility of users based on item ratings.

Thus, we propose a hybrid collaborative filtering algorithm by using double neighbor selection for addressing these problems. There are contributions: One is to improve the quality of neighbors by using user similarity and trust between users as the basis of neighbor selection. The second is to use the similarity between user attributes to solve the problems that users rating data is extremely sparse or cold start in some recommendation system.

## 2 Related Works

In order to improve the accuracy of the recommendation and address how to deal with sparse data and cold start in massive data, Researchers have developed various methods to overcome these problems and given some high quality recommendations. For example, in order to solve the cold start problem of the project, the literature [2] proposed a recommendation algorithm which mainly combines the user time information with the attributes of the item itself and the label information to obtain a personalized prediction score. When the item is cold-started, the item rating can be obtained directly from the item information. However, this method improved the novelty of the algorithm, the accuracy is still slightly lacking. The literature [3] proposed a new collaborative filtering algorithm to mitigate the impact of user cold start in prediction. The literature [4] proposed an attribute-to-feature mapping method to solve the cold start problem in collaborative filtering. The literature [5] proposed a combined recommendation method to solve the sparse data and cold start problems, but the algorithm only produces good recommendations for some and sparse data and it does not achieve an improved effect for a complete cold start. Moreover, the recommendation accuracy for the target user is not high. The literature [6] proposed two recommendation models to solve the complete cold start and incomplete cold start problems for new items, which are based on a framework of tightly coupled CF approach and deep learning neural network. The literature [7] proposed a collective Bayesian Poisson factorization (CBPF) model for handling the new problem of cold-start local event recommendation in Event-based social networks (EBSNs).

The literature [8] proposed a collaborative filtering recommendation algorithm based on multi-social trust. The article quantifies the credibility, reliability and self-awareness in social science to form trust, which improves the accuracy of recommendation and recall rate, although the performance of the algorithm has been reduced. The literature [9] proposed a combination recommendation algorithm based on clustering and collaborative filtering. This algorithm reduces the sparsity of the original matrix by clustering and improves the recommendation accuracy. The literature [10] proposed a general collaborative social ranking model to rank the latent features of users extracted from rating data based on the social context of users. The literature [11] proposed an improved clustering-based collaborative filtering recommendation algorithm. The literature [12] proposed a collaborative filtering algorithm based on the double neighbor selection strategy. Though this algorithm can improve the

recommendation accuracy when the data reaches a certain sparsity, almost half of the users' neighbors cannot reach the expected number in the experiment.

In summary, we propose a combined recommendation algorithm based on collaborative filtering of double neighbor selection strategy. Firstly, the target users are divided into two categories according to the maximum number of neighbors of the target users. Secondly, different recommendation methods are adopted for these two types of users to achieve good recommendation results.

### 3 Hybrid Collaborative Filtering Recommendation Algorithm

#### 3.1 Dynamic Selection of User Interest Similarity Set

In the traditional collaborative filtering system, the user ratings is a set of users consisting of  $m$  users  $U = \{u_1, u_2, \dots, u_m\}$  and the item ratings is set of items consisting of  $n$  items  $T = \{t_1, t_2, \dots, t_n\}$  which is composed of the rating matrix  $R$  of  $m \times n$  order, where  $R_{ij}$  represents the score value of the user  $u_i$  for  $t_j$ , and if the user  $u_i$  has no evaluation for  $t_j$ , the  $R_{i,j} = 0$ .

In the recommendation system, if you want to, how to choose high-quality neighbors is the key to ensuring high recommendation quality. In general, we select neighbors through KNN (K-Nearest Neighbors) [13] in the traditional collaborative filtering algorithm and some of them contain a few users with minor similarity to the target users that leads to the recommendation system accuracy is inefficient. Therefore, in order to reduce the impact of target users' predictions on some their neighbors due to interest bias. The existing work is to define similarity threshold to filter out higher quality neighbors. However, defining a fixed threshold in different environments makes the algorithm less scalable. Thus, Dongyan et al. [12] proposed a method towards dynamic selection of the user's interest similarity set for the target user. The method may set the similarity threshold according to the similarity between the target user  $u_i$  and its neighbor candidate users.

Before getting a similar set of interests for the target user and the target item, we first need to obtain its candidate neighbor set. Let  $C$  denotes candidate neighbor set, Given a historical rating matrix  $R$ , the current target user  $u_i \in U$  and the current target item  $t_j \in T$ , the  $R_{ij} = 0$  at this time. If exists  $u_k \in U$  that makes  $R_{kj} \neq 0$ , we can see that  $u_k$  is the candidate neighbor of the current target user  $u_i$  and the current target item  $t_j$ . Then the candidate neighbor set  $C_j(u_i)$  of  $u_i$  on  $t_j$  can be expressed as follows:

$$C_j(u_i) = \{u_k | R_{ij} = 0 \wedge R_{kj} \neq 0, u_i \in U, t_j \in T\} \quad (1)$$

We regard  $u_k \in C_j(u_i)$  as the recommended neighbor of the target user  $u_i$  corresponding to the target item  $t_j$ , using the Pearson correlation coefficient formula to calculate the similarity between each candidate neighbor and the target user. Let  $CI$  be the common item rating set,  $CI_{ik}$  denotes the set of items which had been rated by user  $u_k$  and  $u_i$ , the similarity calculation between  $u_k$  and  $u_i$  is given by

$$sim(i, k) = \frac{\sum_{t_x \in CI_{ik}} (R_{i,x} - \bar{R}_i)(R_{k,x} - \bar{R}_k)}{\sqrt{\sum_{t_x \in CI_{ik}} (R_{i,x} - \bar{R}_i)^2} \sqrt{\sum_{t_x \in CI_{ik}} (R_{k,x} - \bar{R}_k)^2}} \quad (2)$$

where  $sim(i, k)$  represents the similarity between user  $u_i$  and candidate neighbor user  $u_k$ ;  $R_{i,x}$ ,  $R_{k,x}$  respectively represent the ratings of user  $u_i$  and user  $u_k$  for item  $t_x$ ;  $\bar{R}_i$  and  $\bar{R}_k$  represent the average scores of users  $u_i$  and  $u_k$  on the common item rating set respectively.

For the traditional recommendation algorithm, it is generally necessary to select some neighbors with higher similarity by setting the threshold. In order to improve the scalability of the recommendation algorithm, this paper measures it by the similarity between the target user  $u_i$  and the recommended user  $u_k$ . Its calculation formula is as follows

$$V_{sim} = \frac{\sum_{k=1}^x sim(i, k)}{x}, x = |C_j(u_i)| \quad (3)$$

If  $sim(i, k) \geq V_{sim}$ ,  $u_k$  is a similar user of the target user  $u_i$ . We set the interest similarity set to  $S$ , and then the similar interest set of the target user  $u_i$  corresponding to the target item  $t_j$  can be expressed as  $S_j(u_i)$

$$S_j(u_i) = \{u_k | sim(i, k) > V_{sim}, i \neq k, u_k \in C_j(u_i)\} \quad (4)$$

### 3.2 The Calculation of Trust Degree Between Users

We also use the trust degree between users as one of the criteria for measuring neighbors to improve the accuracy and reliability of the recommendation algorithm.

The  $u_k \in S_j(u_i)$  is taken as the recommended target user of user  $u_i$  and target item  $t_j$ . Referring to the literature [14], the target user  $u_i$  can be predicted based on the following formula

$$P_{i,j} = \bar{R}_i + \frac{(R_{k,j} - \bar{R}_k) \times sim(i, k)}{|sim(i, k)|} \quad (5)$$

Where  $P_{i,j}$  indicates that the user  $u_i$  is based on the predicted score of the input user  $u_k$  on the item  $t_j$ ;  $R_{k,j}$  represents the score of the user  $u_k$  on the item  $t_j$ ;  $\bar{R}_i$  and  $\bar{R}_k$  respectively represent the average score of  $u_i$  and  $u_k$ ;  $sim(i, k)$  represents the similarity between user  $u_i$  and user  $u_k$ .

Based on the deviation of the predicted score  $P_{i,j}$  from the actual score  $R_{i,j}$ , we can calculate the predictive power of the user  $u_k$  for the target user  $u_i$ , which is measured as follows:

$$sat_j(i, k) = \begin{cases} 1, & |P_{i,j} - R_{i,j}| < \varepsilon \\ 0, & \text{else} \end{cases} \tag{6}$$

Where  $sat_j(i, k)$  is the estimated capability value of the user  $u_k$  predicting the user  $u_i$  score, and  $P_{i,j}$  and  $R_{i,j}$  respectively represent the predicted score and the actual score based on the (5) formula.  $\varepsilon$  is a constant, which can be randomly selected. In this paper,  $\varepsilon = 1.2$  is selected by reference [12].

Through  $sat_j(i, k)$  of the formula (6), the calculation formula of the trust degree between the user  $u_i$  and the user  $u_k$  is as follows

$$T(i, k) = \frac{\sum_{x=1}^{|CI_{ik}|} sat_j(i, k)}{|CI_{ik}|} \tag{7}$$

### 3.3 Calculating Similarity Between Users Using User Attributes

In traditional collaborative filtering, the recommendation accuracy and the recall rate decreased owing to the cold start or sparse data. For these reasons, we utilize the user attribute similarity [15] to find the nearest neighbor of the target user and consider the similar user with higher similarity as the recommended user. In this way, we could predict the target user’s rating of the target item based on the recommended user’s rating of the target item.

In general, user attributes include the user’s gender, age, occupation, etc. We can classify users by these attributes and assign specific values. For example, gender can be divided into male and female, we could set “male = 1” and “female = 0”; we segment the age with numerical attributes, for example, the age can be divided by a period of 10 years, so that 0 to 10 years old is 1, 10–20 years old is 2 and so on, other attributes are also assigned in this method. Thus, We can obtain the item attribute matrix *Attr* by processing these raw data information, Table 1 depicts an example user–attribute rating matrix.

**Table 1.** A sample user-attribute matrix

	Gender	Age	Profession
User1	0	1	4
User2	1	5	3
User3	1	3	6
User4	0	1	4
User5	1	2	8

We can obviously represent the feature attribute vector of the user  $u_i$  as  $Attr_{u_i} = \{a_{u_{i1}}, a_{u_{i2}}, \dots, a_{u_{in}}\}$ , if the target user  $u_i$  and the user  $u_k$  has the same  $r$  attribute values, we let  $|a_{u_{ir}} \cap a_{u_{kr}}| = 1$ , otherwise  $|a_{u_{ir}} \cap a_{u_{kr}}| = 0$ . Then users  $u_i$  and  $u_k$  can be measured by the following formula.

$$sim_{attr}(i, k) = \alpha \times |a_{u_{i1}} \cap a_{u_{k1}}| + \beta \times |a_{u_{i2}} \cap a_{u_{k2}}| + \dots + \delta \times |a_{u_{in}} \cap a_{u_{kn}}| \quad (8)$$

Where  $\alpha$ ,  $\beta$  and  $\delta$  are weighting factors, and  $\alpha + \beta + \dots + \delta = 1$ .

Let  $N$  as the nearest neighbor set selected based on the user attribute, the  $N(u_i)$  can be expressed as the nearest neighbor set of the current target user  $u_i$ .

### 3.4 Prediction Computation

In the recommendation system, while many scholars recommend items to the target users, they might add some elements improving the quality of the target users' neighbors as selecting neighbors to improve the recommendation accuracy and reduce the dimension of matrix to improve the efficiency of the algorithm and solve the problem of sparseness of rating matrix. However, in terms of adding elements or reducing matrix dimensions, the following two major problems arise when recommending.

- If we only add some elements that can improve the quality of the neighbors to improve the accuracy of the algorithms. This not only makes the performance of the algorithm decrease, but also makes some users with sparse scoring unable to find a suitable neighbor. Because the neighbors of some data-sparse users cannot meet the requirements of high-quality neighbors. At the same time, the more elements that are added, the longer the algorithm takes, which greatly reduces the efficiency of the algorithm and the recall rate.
- If the matrix dimension reduction method or some user attributes are applied to recommend, the algorithm efficiency and the recall rate can be improved, but the recommended accuracy is greatly reduced.

Therefore, we combine the neighbors selected by the dynamic neighbor with the trust factor which is more important to the recommendation algorithm to improve the neighbor quality. Moreover, for some users with sparse data or cold start, they are recommended based on the similarity of user attributes and then a hybrid collaborative filtering algorithm based on double neighbor selection is proposed, and named HCF-DNS. The core ideas of HCF-DNS are as follows

1. Selecting a neighbor candidate set  $C_j(u_i)$  of the target user  $u_i$  for the target item  $t_j$ ;
2. Obtaining a dynamic similar interest set  $S_j(u_i)$  of the target user  $u_i$  based on the target item  $t_j$  according to the method of dynamically selecting a neighbor user in Sect. 3.2;
3. If the similar interest set  $S_j(u_i)$  of the target user  $u_i$  is an empty set, the predicted score of the current target user  $u_i$  on the item  $t_j$  is calculated based on the similarity of the user attributes. The predicted score formula is as follows

$$P_{i,j} = \frac{\sum_{k \in N(u_i)} sim_{attr}(i, k) \times R_{k,j}}{\sum_{k \in N(u_i)} |sim_{attr}(i, k)|} \quad (9)$$

If a cold start item occurs, the target user's rating for the cold start item is set to 3;

4. If the similar interest set  $S_j(u_i)$  of the target user  $u_i$  is not an empty set, the trust degree of each of the target user  $u_i$  and the user similar interest set  $S_j(u_i)$  is obtained in using the inter-user trust degree calculation method in Sect. 3.3. And select the Top-N users with the most trust as the trusted neighbor user set of the target user  $u_i$ ;
5. Calculating the score of the target user  $u_i$  on the target item  $t_j$  based on the rating information of the neighbor of the trusted user and using the following equation.

$$P_{i,j} = \bar{R}_i + \frac{\sum_{u_k \in T(u_i)} (R_{k,j} - \bar{R}_k) \times T(i,k)}{\sum_{u_k \in T(u_i)} |T(i,k)|} \quad (10)$$

According to the above algorithm idea, the hybrid collaborative filtering algorithm based on double neighbor selection is given as follows

---

**Algorithm:** HCF-DNS

**Input:** Score matrix  $R$ , user-attribute matrix  $Attr$ , target user  $u_i$ , target item  $t_j$

**Output:** The target user  $u_i$  predicts the score of the target item  $t_j$ ,  $P_{i,j}$

---

Begin:

$T(u_i) \leftarrow \emptyset$ ,  $N(u_i) \leftarrow \emptyset$

$S_j(u_i) = \{u_k | sim(i,k) > V_{sim}, i \neq k, u_k \in C_j(u_i)\}$

if  $S_j(u_i) = \emptyset$

  for each  $u_k \neq u_i$

    utilize equation (8) to obtain similarity with  $u_i$   
  end for

  sort the similarity of  $u_i$  and each  $u_k$  to get  $N(u_i)$   
  select Top-N users with the highest similarity in

$N(u_i)$

  utilize the equation (9) to get the predicted rating  $P_{i,j}$  of the target user  $u_i$

else

  for each  $u_k \neq u_i$

    utilize equation (8) to obtain trust degree with  $u_i$   
  end for

$T(u_i) \leftarrow$  Top-N users with the most trust degree

  utilize the equation (10) to get the predicted rating

$P_{i,j}$

return  $P_{i,j}$

End

---

## 4 Experiment and Results Analysis

### 4.1 Test Data Set

We adopt the dataset (<https://movielens.umn.edu/>) provided by the MovieLens [16]. In this data set, users rate on the movies they have seen with a rating range of 1 to 5. The higher the score, the higher the user's satisfaction with the movie. The data set also provides some basic information about the user, such as the user's gender, age and occupation. In this experiment, this information is collected as user attribute information. In the course of the experiment, we will collect 3 attributes and the weighting factors were set to  $\alpha = 0.45$  (gender weighting factor),  $\beta = 0.35$  (age weighting factor),  $\delta = 0.2$  (occupation weighting factor) after referencing the literature [17]. There are 943 users in the dataset with 100,000 ratings for 1,682 movies and the its sparsity of approximately 93.7%.

In the experiment, we divided the data set into training sets and test sets according to the ratio of 80% and 20%.

### 4.2 Performance Evaluation Index

We make use of the mean absolute error (MAE) [1, 18], root mean square error (RMSE) [1, 18] and recall rate (RL) [1] to evaluate the prediction accuracy and prediction ability of the recommendation algorithm. The MAE and the RMSE are important parameters for evaluating the accuracy of the recommendation algorithm and the RL is the precision of the evaluation recommendation system. The MAE is obtained by the deviation between the predicted score and the actual score. The lower the MAE value, the higher the accuracy of the recommendation algorithm; the RMSE with high sensitivity to the error is the square root of the ratio of the squared deviation of the predicted rating to the actual rating and the predicted number of times. The lower the RMSE, the more accurate the prediction score. The RL is also called the recovery rate, which is used to measure the recall and precision of the algorithm. The measurement equation is as follows.

$$\text{MAE} = \frac{\sum_{k=1}^n |P_k - R_k|}{n} \quad (11)$$

Where  $P_k$  represents the rating prediction value,  $R_k$  represents the real evaluation value and  $n$  represents the number of evaluations.

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^n (P_k - R_k)^2}{n}} \quad (12)$$

Among them, the definition of the variable appearing in the formula is the same as the Eq. (11).

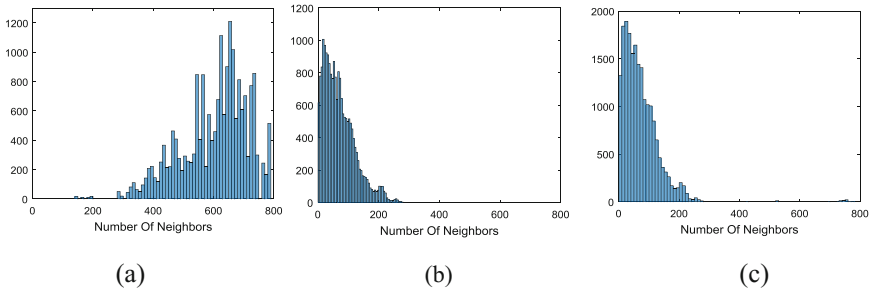


$$RL = \frac{m}{n} \tag{13}$$

Where  $m$  represents the number of ratings that can be predicted by the algorithm, and  $n$  represents the number of ratings to be tested in the test set.

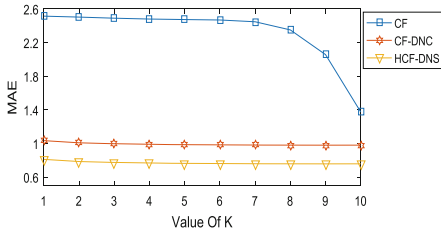
### 4.3 Experimental Method

Since the algorithm proposed in this paper is for various users and the number of corresponding neighbors is relatively scattered when selecting the similar interest sets of the target users, in order to make the comparison between the proposed algorithm and other algorithms more obvious during the test. We select  $1/10, 2/10, \dots, k/10, 1$  of the number of neighbors to test. The following is the distribution of similar neighbors of the target users in the test set. The following is the distribution of similar neighbors of the target users in the test set one.

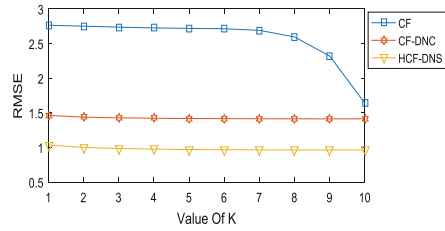


**Fig. 1.** The distribution of neighbor numbers for three algorithms

In the above figures, Fig. 1(a) is the neighbor distribution of the target user corresponding to the target item in using the CF algorithm. From this figure, we can see that the number of selected neighbors is concentrated in the interval of 400 to 800. Thus, the neighbors are not filtered when making predictions. Although the number of neighbors is large, the accuracy is not high; Fig. 1(b) is the number of neighbors selected after using the CF-DNC algorithm. As can be seen from the figure, the number of neighbors is mostly distributed in the interval of 0 to 200 and there are target users with zero neighbors. Although the quality of neighbors has improved, the cold start problem is still inevitable when we use the measures in [20]. Figure 1(c) is the improved algorithm HCF-DNS algorithm, the distribution of neighbors is also 0 to 200 but it can be seen that the algorithm has screened the number of neighbors for high quality after comparing with the Fig. 1(a). At the same time, combined with the recall rate of Table 2 and Fig. 1(b) the comparison, it can be seen that the cold start problem has also been alleviated.



**Fig. 2.** The results of using MAE



**Fig. 3.** The results of using RMSE

**Table 2.** Comparison of recall rates

	CF	CF-DNC	HCF-DNS
Recall	1	0.91	1

#### 4.4 Experimental Results and Analysis

In order to evaluate the accuracy of the algorithm proposed in this paper, we compare it with the traditional collaborative filtering algorithm (CF) [19] and the previously mentioned double neighbor choosing strategy based on collaborative filtering (CF-DNC) in the same experimental environment, the experimental results are as follows.

It can be seen from Figs. 2 and 3 that the algorithm HCF-DNS is better than CF-DNC and traditional CF. As can be seen from Table 2, the recall rate of the algorithm is relatively high. It can be seen that in the case of extremely sparse data sets, we not only improves the recommendation accuracy, but also improves the recall rate.

The reason why our recall rate is higher than other experiments is that we refer to the literature [20] to do some processing on some sparse data of CF and CF-DNC during the experiment. And after processing, we find that our proposed algorithm still has an advantage in recall rate.

The main reason is that in the recommendation process, we add the trust degree factor, which is more important among users to improve the recommendation accuracy. When the user starts coldly, the user attribute-based recommendation is adopted to avoid the decrease of the recall rate. Thus avoiding the problem of the drop in the recall rate. It can be seen that HCF-DNS proposed in this paper achieves better results than CF and CF-DNC.

## 5 Conclusion and Further Work

With the wide application of recommendation technology in e-commerce, both the accuracy of the recommendation system and the recall rate are the main research directions to improve QoS of recommendation. In this paper, a hybrid collaborative filtering algorithm based on double neighbor selection, which named HCF-DNS, is proposed for addressing these two problems.

The contributions of proposed algorithm are that taking the trust degree between users as the basis of the target user's neighbors selection, and mixing the demographic-based recommendation algorithm to prevent the cold start problem. The proposed algorithm has effectively addressed the problem of user cold start in traditional collaborative filtering and improved Recommended accuracy.

However, the demographic recommendation algorithm is a coarse-grained recommendation algorithm. The accuracy of the recommended algorithm is not high when solving the cold start. So how to design a more effective recommendation algorithm, making the recommendation efficiency of cold start users higher is the next research work of this paper.

**Acknowledgments.** The paper is supported in part by the National Natural Science Foundation of China under Grant No. 61672022, and Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University under Grant No. XXKZD1604.

## References

1. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems: Introduction and Challenges. *Recommender Systems Handbook*, pp. 1–34. Springer, Boston (2015). <https://doi.org/10.1007/978-0-387-85820-3>
2. Yu, H., Li, J.: Algorithm to solve the cold-start problem in new item recommendations. *J. Softw.* **26**(6), 1395–1408 (2015)
3. Bobadilla, J., et al.: A collaborative filtering approach to mitigate the new user cold start problem. *Knowl. Based Syst.* **26**, 225–238 (2012)
4. Gantner, Z., et al.: Learning attribute-to-feature mappings for cold-start recommendations. In: 2010 IEEE 10th International Conference on Data Mining (ICDM). IEEE (2010)
5. Sobhanam, H., Mariappan, A.K.: Addressing cold start problem in recommender systems using association rules and clustering technique. In: International Conference on Computer Communication and Informatics (ICCCI) (2013)
6. Wei, J., et al.: Collaborative filtering and deep learning based recommendation system for cold start items. *Expert. Syst. Appl.* **69**, 29–39 (2017)
7. Zhang, W., Wang, J.: A collective bayesian poisson factorization model for cold-start local event recommendation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2015)
8. Ruiqin, W., et al.: A collaborative filtering recommendation algorithm based on multiple social trusts. *J. Comput. Res. Dev.* **53**(6), 1389–1399 (2016)
9. Napoleon, D., Ganga Lakshmi, P.: An efficient K-Means clustering algorithm for reducing time complexity using uniform distribution data points. In: *Trendz in Information Sciences & Computing (TISC)*. IEEE (2010)
10. Forsati, R., et al.: Pushtrust: an efficient recommendation algorithm by leveraging trust and distrust relations. In: Proceedings of the 9th ACM Conference on Recommender Systems. ACM (2015)
11. Xiaojun, L.: An improved clustering-based collaborative filtering recommendation algorithm. *Clust. Comput.* **20**(2), 1281–1288 (2017)
12. Dongyan, J., et al.: A collaborative filtering recommendation algorithm based on double neighbor choosing strategy. *J. Comput. Res. Dev.* **50**(5), 1076–1084 (2013)

13. Herlocker, J.L., et al.: An algorithmic framework for performing collaborative filtering. In: ACM SIGIR Forum, vol. 51, no. 2. ACM (2017)
14. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: Proceedings of the 10th International Conference on Intelligent User Interfaces. ACM (2005)
15. Raad, E., Chbeir, R., Dipanda, A.: User profile matching in social networks. In: 2010 13th International Conference on Network-Based Information Systems (NBIS). IEEE (2010)
16. Miller, B.N., et al.: MovieLens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th International Conference on Intelligent User Interfaces. ACM (2003)
17. Rydén, F., et al.: Using kinect and a haptic interface for implementation of real-time virtual fixtures. In: Proceedings of the 2nd Workshop on RGB-D: Advanced Reasoning with Depth Cameras (in conjunction with RSS 2011) (2011)
18. Karafyllis, I.: Finite-time global stabilization by means of time-varying distributed delay feedback. *SIAM J. Control Optim.* **45**(1), 320–342 (2006)
19. Herlocker, J.L., et al.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. (TOIS)* **22**(1), 5–53 (2004)
20. Schein, A.I., et al.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2002)