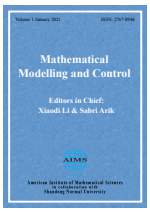
Mathematical Modelling and Control

http://www.aimspress.com/journal/mmc



MMC, 3(1): 39–49
DOI:10.3934/mmc.2023004
Received: 03 September 2022
Revised: 26 December 2022
Accepted: 06 January 2023
Published: 15 February 2023

Research article

Improved collaborative filtering personalized recommendation algorithm based on k-means clustering and weighted similarity on the reduced item space

Jiaquan Huang, Zhen Jia*and Peng Zuo

College of Science, Guilin University of Technology, Guilin, 541004, China

* **Correspondence:** Email: jjjzzz0@163.com.

Abstract: Collaborative filtering (CF) algorithm is one of the most widely used recommendation algorithms in recommender systems. However, there is a data sparsity problem in the traditional CF algorithm, which may reduce the recommended efficiency of recommender systems. This paper proposes an improved collaborative filtering personalized recommendation (ICF) algorithm, which can effectively improve the data sparsity problem by reducing item space. By using the k-means clustering method to secondarily extract the similarity information, ICF algorithm can obtain the similarity information of users more accurately, thus improving the accuracy of recommender systems. The experiments using MovieLens and Netflix data set show that the ICF algorithm has a significant improvement in the accuracy and quality of recommendation.

Keywords: recommender systems; personal recommendation algorithm; k-means clustering; bipartite network; collaborative filtering

1. Introduction

With the rapid development of network technology, the amount of information that users can obtain online has increased dramatically, resulting in information overload problems [1, 2]. As an important player in the field of personalized services, recommender systems can effectively alleviate the information overload problem by analyzing the relationship between users, and it provides users with information and services of interest based on their preferences or historical records [3]. It offers a wide range of applications and commercial value in the domains of e-commerce and precision advertising.

Recommender systems generally contain three modules: user, recommendation object and recommendation algorithm, of which the recommendation algorithm is the most important part, and the core part of the system. The existing recommender systems can be divided into three categories according to the degree of personalization [4]. The first is a non-personalized recommendation system, which recommends the same items for each user according to various item rankings, and its recommendation algorithm only applies some simple statistical methods, such as global ranking method (GRM [5]). The second is a semi-personal recommendation system, which recommends items that users have browsed, mainly based on association rule analysis [6]. The last is a personalized recommendation system, which provides users with items of interest by analyzing the preferences or histories of user. The classical recommendation algorithms, include content-based recommendation, collaborative filtering recommendation and hybrid recommendation [7].

Collaborative filtering (CF) algorithm was first proposed by Greg Linden et al. in 2003, and has become one of the most widely used algorithms in the recommendation system [8]. The basic idea is to recommend items

of interest to the target user through the browsing or rating records of the user. Subsequently, scholars have proposed some new recommendation algorithms based on the idea of collaborative filtering [9–16]. Xia et al. [9] service recommendation as a multi-objective optimization problem, and propose two improved ranking prediction and recommendation algorithms, taking accuracy and diversity into consideration. Yu et al. [10] proposed a context-enhanced deep neural collaborative filtering (CDNC) item recommendation model, which utilized the textual features of items and collaborative features (user ratings) to alleviate the cold-start problem, and provided recommendations to users. A single-objective hybrid evolutionary algorithm was proposed by Touraj et al. [11] for item clustering in offline collaborative filtering recommender systems, and the Genetic Algorithm (GA) and Gravitational Emulation Local Search (GELS) algorithm were used to create a stable algorithm for more appropriate clustering of data in recommender systems, which then improve the accuracy of the system. Minh et al. [12] proposed a context-aware recommendation method that is less sensitive to the data sparsity. The method exploited the transferability of interactions between users and items on the item graph to enhance the connection of such direct interactions, thus reducing the negative impact of sparse data, and giving relevant recommendations. Feng et al. [13] suggested a multi-factor similarity measure based on matrix decomposition under the CF algorithm, as well as a fusion approach of multi-factor similarity and global rating information to increase the robustness of the recommendation system and the prediction accuracy of sparse data. Lv et al. [14] proposed a new attention-based item collaborative filtering (AICF) model that used three distinct attention processes to estimate the weights of historical things with which users had interacted, and AICF model had outstanding recommendation performance on sparse data. Achraf et al. [15] proposed a new similarity measure method. The method transformed some intuitive and qualitative conditions that the similarity measure should satisfy into a related set of mathematical equations, and its kernel function of the similarity measure was obtained by solving the set of equations. Alhijawi et al. [16] proposed a prediction model that used a genetic-based prediction model (INH-BP), and a suitable heuristic search algorithm to effectively alleviate the two main issues (cold start and sparsity) of recommender systems.

At present, there are two main types of CF algorithm: nearest neighbor-based and model-based algorithms [17]. The idea of the nearest neighbor-based collaborative filtering is to give a recommendation list by calculating the similarity between users or items; the idea of model-based CF [18] is to give a recommendation list by training a model with information about the selected items. However, CF algorithm has some limitations in practical application, especially in the environment of sparse data, where its recommendation accuracy will be greatly reduced. How to improve the recommendation effect of CF algorithm in sparse data environment? That is the focus of this paper.

In this paper, we propose a new improved CF algorithm (called ICF algorithm). In the reduced item space, a weighted similarity model is established based on k-means clustering, and the PSO algorithm is applied to further optimize the model. With the model, ICF algorithm can extract the similarity information twice between users, which will get more information about users, and obtain the final recommendation lists. The ICF algorithm not only integrates the local information and global information of the data, but also reduces the frequency of applying CF algorithm and improves the recommendation efficiency of the recommendation system. Finally, the effectiveness of the ICF algorithm is verified by the experiments on benchmark data (MovieLens and Netflix data set).

The rest is organized as the following: The bipartite network and the calculation of similarity are briefly introduced in Section 2. In Section 3, a detailed description of the ICF algorithm is given. In section 4, a series of experiments are conducted to compare ICF algorithm with several algorithms. Finally, conclusions are discussed in section 5.

2. Preliminaries

With the development of network technology and the increase of network data, the information overload problem, when people accessing information on the network, has caused a lot of troubles. Recommender systems have been

a popular research area for scholars to solve the information overload problem. Nowadays, clustering methods have also received wide attention in the research of recommendation systems. Singh M et al. proposed a biclustering based collaborative filtering (BBCF) recommendation algorithm [19]. With enabling simultaneous clustering of users and items based on the partial matching of users' preferences on items, the BBCF systems has better performance compared to the state-of-the-art rating prediction approaches. Kant Set et al. proposed a unique centroid selection approach for k-means clustering algorithm for collaborative filtering [20], and the accuracy fo recommendation systems was effectively improved. Chen J et al. proposed a dynamic evolutionary clustering approach based on time weight and latent attributes for collaborative filtering recommendation [21], which improved the recommendation accuracy. Vara et al. used a k-means clustering algorithm to improve the effectiveness of the results recommended by journal recommender system, which resulted in a better recommendation performance [22]. In order to solve the data sparsity problem, this paper proposes an improved collaborative filtering personalized recommendation algorithm (ICF algorithm) based on k-means clustering and weighted similarity on reducing item space. Compare to the other algorithms (CF, GRM, COSCF and JaccardCF algorithm), the recommendation accuracy of ICF is imprived with the k-means clustering to gain more user similarity.

2.1. Basic data and bipartite network

In general, the experimental data mainly includes three parts: user set, item set and user-item interaction relationship. The user-item interaction relationship refers to the historical connection between the user and the item, which is generally represented by 0-1 variables. For example, if a customer has bought or bookmarked an item on an e-commerce platform, the association between the user and the item is represented by 1; otherwise it is represented by 0. A bipartite network can be built by the above-mentioned interaction links between users and items in a recommender system. Suppose the recommender system has n users (denoted by u_i, $i = 1, 2, \ldots, n$) and m items (denoted by o_j, $j = 1, 2, \ldots, m$), then their corresponding bipartite network is described by the adjacency matrix A:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \tag{2.1}$$

where $a_{ij} = 1$ or $0 (i = 1, 2, \ldots, n, j = 1, 2, \ldots, m)$. $a_{ij} = 1$, which means u_i has selected o_j; otherwise $a_{ij} = 0$. We will use the adjacency matrix A as the basis for constructing the recommendation algorithm.

The similarity between u_i and u_j can be measured by the following methods.

(1) Similare to pearson coefficient:

$$s_{ij} = \frac{\sum_{l=1}^{m} a_{il} a_{jl}}{\min(k(u_i), k(u_j))}, \tag{2.2}$$

where a_{il} is the elements of matrix A, $k(u_i) = \sum_{l=1}^{m} a_{il}$ is the degree (the number of items selected by u_i) of u_i, the similarity between u_i and u_i is 1, and $s_{ij} \in [0, 1]$.

(2) Cosine similarity [23]

$$cos_{ij} = \frac{\sum_{l=1}^{m} a_{il} a_{jl}}{\sqrt{\sum_{l=1}^{m} a_{il}^2} \sqrt{\sum_{l=1}^{m} a_{jl}^2}}, \tag{2.3}$$

(3) Jaccard similarity [24]

$$Jaccard_{ij} = \frac{\sum_{l=1}^{m} a_{il} a_{jl}}{\sum_{l=1}^{m} a_{il}^2 + \sum_{l=1}^{m} a_{jl}^2 - \sum_{l=1}^{m} a_{il} a_{jl}}, \tag{2.4}$$

2.2. Accuracy evaluation index

In the specific application of recommendation algorithms, some evaluation metrics need to be used to measure the performance of the algorithm. Generally, we divide the data set randomly into two parts: training set and test set in experiments. With the data in the training set, we obtain the prediction scores (of items that are not selected by users) and the recommendation lists, and we check the accuracy of the prediction results with the data in the test set. In recommender systems, the accuracy of prediction can be measured from different perspectives, and the ranking accuracy, precision, recall, novelty and diversity are commonly used to be the evaluation metrics [23, 24]. The higher precision and recall are expected in the better recommendation effect, and the lower ranking accuracy,

novelty and diversity denote the excellent recommendation results.

(1) Sorting accuracy

The sorting accuracy plays a critical role on the recommendation system that requires strict ordering. For example, if the favorite item of a user is listed third in the recommendation list, whereas the least favorite item is ranked first, the satisfaction of the user with the recommendation system will decrease. The sorting accuracy is calculated as follows:

$$r_{ij} = \frac{l_{ij}}{L_i}. \tag{2.5}$$

Suppose there is a test set containing users, items and ratings, where r_{ij} denotes the ranking accuracy of o_j selected by u_i in the test set, l_{ij} denotes the ranking that u_i selected o_j in the recommendation list (the length of the list is L_i). The ranking accuracy $\overline{r}$ of the system is obtained by averaging the ranking accuracies of all users. The calculation formula of $\overline{r}$ is as follows:

$$\overline{r} = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{M_i} r_{ij}/M_i)}{n}, \tag{2.6}$$

where M_i is the number of in the test set items selected by u_i.

(2) Precision

Precision $p(u_i)$ reflects the probability that whether u_i is interested in the items recommended by the recommendation system. All items that need to be recommended are ranked according to their prediction scores, and the system will recommend the top L items to u_i. The calculation formula is as follows:

$$p(u_i) = \frac{\Gamma(u_i) \cap T(u_i)}{L}, \tag{2.7}$$

where $\Gamma(u_i)$ denotes the set of recommended items of u_i, L is the length of the recommendation list, and $T(u_i)$ denotes the set of items in the test set. The system average precision is defined as follows:

$$\overline{p} = \frac{\sum_{i=1}^{n} p(u_i)}{n}. \tag{2.8}$$

(3) Recall

Recall $Recall(u_i)$ reflects the probability that the likelihood of an item liked by u_i is recommended, and is defined as the ratio of items liked by users in the recommendation list to all items liked by users in the test set. It is calculated as follows:

$$Recall(u_i) = \frac{|\Gamma(u_i)| \cap |T(u_i)|}{|T(u_i)|}, \tag{2.9}$$

where $|T(u_i)|$ denotes the number of items selected by u_i in test set.

The average recall rate ($\overline{Recall}$) of the system is defined as:

$$\overline{Recall} = \frac{\sum_{i=1}^{n} Re(u_i)}{n}. \tag{2.10}$$

(4) Novelty

$$Novelty = \frac{1}{nL}\sum_{i=1}^{n}\sum_{l=1}^{L} k(o_i), \tag{2.11}$$

where $k(o_i)$ is the degree of o_i.

(5) Diversity

$$Diversity = \frac{1}{n(n-1)}\sum_{i\neq j} 1 - \frac{\Gamma(u_i) \cap \Gamma(u_j)}{L}. \tag{2.12}$$

3. ICF algorithm

In order to improve the recommendation effect of CF algorithm in sparse data problem, the ICF algorithm uses k-means to cluster user information, and builds a weighted similarity model on reducing item space, and, finally, applies the particle swarm algorithm (PSO) to optimize the model. Based on the model, the algorithm can extract the similarity information of users twice, and obtain the final recommendation lists. The steps of ICF algorithm are as follows:

Step 1: Reduce item space

First, the items are categorized, and the categorization methods are different in different environments. For example, if the items are commodities, they can be categorized by usage or by the labels of the items, which reduces the item space, and transforms the bipartite network of the underlying data into a weighted bipartite network of the set of user-item attributes. Generally, we convert the adjacency matrix A (in (2.1)) of the bipartite network to the corresponding adjacency matrix B (of the reduced space weighted bipartite network). Matrix B is defined as follows:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1d} \\ b_{21} & b_{22} & \dots & b_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{2d} \end{bmatrix}, \tag{3.1}$$

where $b_{ij} = \sum_{k \in N_j} a_{ik}$ indicates that the number of items that u_i has selected in j category items, where $N_j (j = 1, 2, \dots, d)$ denotes the set of items belonging to the same j category, and d is the number of categories of items. There is an example of reducing item space, as shown in Figure 1. In Figure 1, the left panel shows the bipartite network of users and items, and the right panel shows the bipartite network after reducing item space.

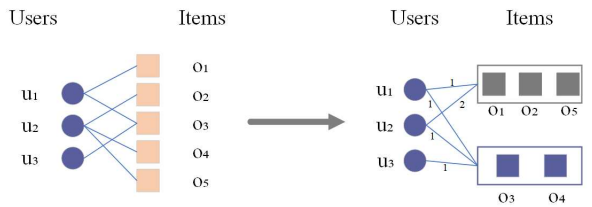


Figure 1. An example of reducing item space.

From Figure 1, we can obtain the adjacency matrix A and matrix B of the bipartite network after reducing item space, as shown in equation (3.2).

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow B = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 0 & 1 \end{bmatrix}. \tag{3.2}$$

The effect of reducing the space is that, on the one hand, it focuses on the interests of users on the item attributes instead of individual items, which is equivalent to refining the preferences of users from a higher perspective. On the other hand, it reduces the item dimension, which not only reduces the computation, but also can effectively solve the situation of algorithm failure due to the sparsity, and it broadens the application of our algorithm.

Step 2: Secondary extraction of user information

In order to measure the similarity between users accurately, we first apply the k-means clustering to classify users by using matrix B (a row in matrix B represents a sample, and a column represents an item category) as the clustering data, and the Euclidean distance as the similarity measure to perform k-means clustering:

(1) First, k samples (elements of k rows of matrix B) are randomly selected as the initializations of k-means, and the set of clusters $C = c_1, c_2, ..., c_k$ are obtained. In order to eliminate the influence of the initial clustering center on the final result, the ICF algorithm will be optimized by using the PSO algorithm.

(2) Second, we modify $C = c_1, c_2, \dots, c_k$. We will calculate the distance between each sample and these clustering primes, and cluster the shortest distance into one class, and update the clustering primes.

(3) Finally, repeat step (2) until no new clusters are generated and all samples are not available at the time of assignment to obtain the final set of clusters $C^* = c_1^*, c_2^*, \dots, c_k^*$. The purpose of clustering is to extract the similarity information of users, again, on the reduced item space, and to cluster users with similar interests into one category.

Step 3: Calculate the weighted similarity

It is generally believed that the similarity between users located in the same cluster is higher than that of users in different clusters, so we recalculate the similarity of users using the clustering information on the reduced item space, and add a weight parameter α to the similarity s_{ij} in (2.2) to obtain the weighted similarity $\widetilde{s}_{ij}$ of u_i and u_j:

$$\widetilde{s}_{ij} = \begin{cases} \alpha s_{ij}, & u_i, u_j \in c_l^* (l = 1, \cdots, k) \\ (1 - \alpha) s_{ij}, & others \end{cases}, \tag{3.3}$$

where s_{ij} is the similarity in (2.2) of u_i and $u_j (i = 1, \cdots, m$, $j = 1, \cdots, n)$, and $\alpha (0.5 < \alpha < 1)$ is the adjustable weight parameter, $s_{ij} \in [0, 1]$.

Step 4: Calculate the prediction score

Using $\widetilde{s}_{ij}$ to calculate prediction score $\widetilde{w}_{ij}$ of o_j relative to u_i:

$$\widetilde{W}_{ij} = \frac{\sum_{l=1, l \neq i}^{n} \widetilde{s}_{il} a_{jl}}{\sum_{l=1, l \neq i}^{n} \widetilde{s}_{il}}, \tag{3.4}$$

where s_{ij} is the weighted similarity of u_i and u_j.

After using k-means clustering, users with high similarity will be clustered into one class, and the recommender system has collected higher quality information of u_i, so it can

recommend more suitable items and provide more effective personalized services for u_i.

Step 5: Optimize k and α

The PSO algorithm is a class of stochastic global optimization techniques that discover optimal regions in complex search spaces through inter-particle interactions, and PSO has the advantage of being simple, easy to implement and powerful. From (3.3), it can be seen that the similarity $\widetilde{s}_{il}$ of u_i and u_j is affected by k and α after clustering, so in order to get the optimal k and α, the particle swarm algorithm (PSO) is used in the experiment to optimize them.

PSO is initialized as a population of random particles (the random solutions are the k and α), and then the optimal solution is found by iteration. In each iteration, the particles update themselves by tracking two extremums (ranking accuracy in (2.5)). The first extremum is the optimal solution found by the particle itself, called the individual extremum p_{best}. The other is the optimal solution currently found by the whole population, called the global extremum g_{best}. The update and iteration equations of PSO are shown below:

$$\begin{cases} v_i = v_i + c_1 r_1(p_{best} - x_i) + c_2 r_2(g_{best} - x_i), \\ x_i = x_i + v_i, \end{cases} \tag{3.5}$$

where v_i is the velocity of the particle, and $x_i = (\alpha_i, k_i)(\alpha_i \in (0.5, 1), k_i \in (2, 120))$ is the current position of the particle, where usually, $c_1 = c_2 = 2$, r_1 and $r_2(r_1, r_2 \in (0, 1))$ are the random numbers. Finally, the optimal parameter is denoted as (k^*, α^*).

Step 6: Calculate the final w^*_{ij} and make recommendations

Bring the optimal (k^*, α^*) obtained in step 5 into (3.3), so that $\widetilde{s}^*_{ij}$ is calculated by the following:

$$\widetilde{s}^*_{ij} = \begin{cases} a^* s_{ij}, \quad u_i, u_j \in c_l(l = 1, \cdots, k^*), \\ (1 - a^*)s_{ij}, \quad others. \end{cases} \tag{3.6}$$

then, the final $\widetilde{w}^*_{ij}$ is calculated:

$$\widetilde{W}^*_{ij} = \frac{\sum_{l=1, l\neq i}^{n} \tilde{s}^*_{il} a_{jl}}{\sum_{l=1, l\neq i}^{n} \tilde{s}^*_{il}}. \tag{3.7}$$

For the u_i, we have calculated its w^*_{ij} for the set of unselected o_j, elements of w^*_{ij} are sorted in descending, and finally, items with the highest w^*_{ij} are recommended to u_i. The ICF algorithm is shown as Algorithm 1.

Algorithm 1 ICF algorithm

Input: Adjacency A
Output: Recommendation list L
$B \Leftarrow FunctionGetB(A)$
$(k^*, \alpha^*) \Leftarrow FunctionPSO(k - means(B, k), \alpha)$
for i = 1 to n **do**
    for i = 1 to m **do**
        $\widetilde{s}^*_{ij} \Leftarrow Function Get\widetilde{s}^*(s_{ij}, \alpha^*)$
    end for
end for
$\widetilde{S}^* \Leftarrow \{\widetilde{s}^*_{ij}\}$
$\widetilde{w}^* \Leftarrow FunctionGetW(\widetilde{S}^*)$
for i = 1 to n **do**
    $L(u_i) \Leftarrow FunctionGetL(\widetilde{w}^*)$
end for

4. Experiments and results analysis

At present, the classical recommendation algorithms are CF and GRM. To test the efficiency of the ICF algorithm, ICF algorithm is compared with CF and GRM algorithm. In addition, there are two other methods in the CF algorithm for calculating similarity, such as Cosine similarity [25] (in equation (2.3)) and Jaccard similarity [26] (in equation (2.4)), respectively, called COSCF and JaccardCF algorithm. Further the COSCF and JaccardCF algorithms will be added to compare with the ICF algorithm. The experiments are intended to address the following questions: (1) How do k and α affect ICF algorithm performance? (2) How does ICF algorithm perform for MovieLens and Netflix data set compare with other algorithms?

4.1. Experimental data set

The MovieLens data was collected through the MovieLens web site (movielens.umn.edu), and it was a common benchmark data set used to test the recommendation algorithm. This data has been cleaned up with users who had less than 20 ratings or did not have complete demographic information being removed from this data set. Movielens data set contained of 100,000 ratings (1-5) from 943 users on 1682 movies. Netflix data included

19,7248 ratings (1-5) from 3000 users on 2779 movies. In general, 90% of the data set are randomly selected as the training set, and the remaining 10% as the test set.

4.2. Experiments and analysis of results

The experiment first considers the effect of k and α in (3.3) on the ranking accuracy, and optimizes them using the PSO, and calculates their optimal values. Then, the accuracy of the ICF algorithm, based the optimal values (k^*, α^*), is compared with the traditional CF algorithm.

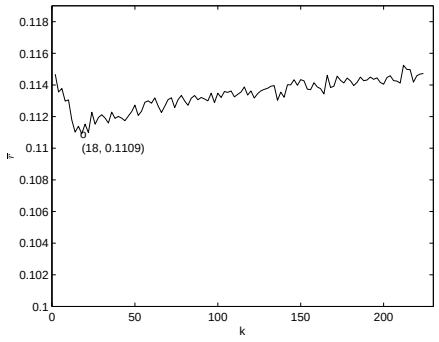
(1) The sensitivity of ICF algorithm performance to k and α.

The parameter k is critical for k-means clustering to cluster the users. If it is small, the extraction of user similarities is not comprehensive enough, and there will be low accuracy of recommendations, and the diversity of recommendations becomes poor. When it is large, it will increase the complexity of the algorithm, and affect the recommendation performance of the recommendation system. In the experiment, different values of parameter k and α are taken for each of the four different evaluation metrics of the recommended system, $L = 100$.

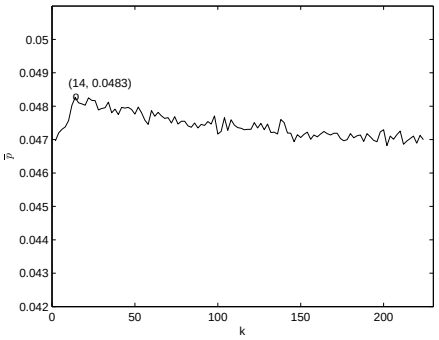
First, we test the effect of the parameter on the accuracy of the ICF algorithm, the experiment takes the parameter $\alpha = 0.8$, and the results are shown in Figure 2.

Figure 2 shows the experimental results of sorting accuracy, precision and recall rate of the ICF algorithm when parameter k is changed. In Figure 2, it can be seen that when parameter $k = 18$, ICF algorithm's average sorting accuracy is the lowest. When $k = 14$, its precision is the highest. When parameter $k = 14$, its recall rate is the highest. When $k > 18$, and as k increases, the average ranking accuracy shows a gradual increase, while the precision shows a gradual decrease, the recall rate shows a gradual decrease. This indicates that the larger k is not always better.

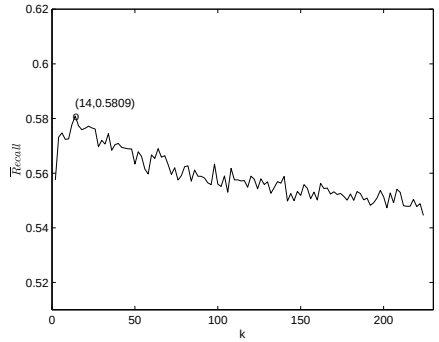
Figure 3 shows the experimental results of sorting accuracy, precision and recall rate of the ICF algorithm when parameter α is changed. As seen in Figure 3, when parameter $\alpha = 0.92$, the average ranking accuracy and precision of the recommendation reach the optimum. When $\alpha = 0.98$, the recall rate of the recommendation is optimal. When $\alpha > 0.92$, the ranking accuracy tends to increase and the precision tends to decrease, which also indicates that larger α is not always better.



(a) Average sorting accuracy



(b) Precision



(c) Recall

Figure 2. Performance comparison based on precision, recall and sorting accuracy with different k.

Figure 4 shows the trend of the average ranking accuracy of the ICF algorithm in the (k, α) plane chromatogram, when

k and α are varied, jointly. It can be seen that the average sorting accuracy in the blue region in Figure 4 is optimal. Finally, the optimal values ($k^* = 8$, and $\alpha^* = 0.92$) are obtained by PSO.

(2) Accuracy comparison of different algorithms

While the optimal values ($k^* = 8$, and $\alpha^* = 0.92$) are obtained, the accuracy of different algorithms (CF, GRM, COSCF and JaccardCF algorithm) are compared on the Movielens and Netflix data set. The results are shown in Table 1 and Table 2.

Table 1 shows the experimental results of ranking accuracy, precision, recall, novelty and diversity of different algorithms. From Table 1, it can be seen that the ranking accuracy, precision, recall, novelty and diversity of the ICF are significantly better than other algorithms on MovieLens data. On the Netflix data, ICF also has the better performance. Table 2 displays the improvements of ICF algorithm compared to other algorithms. For example, relative to CF, the precision of ICF increased by 11.3% on MovieLens data. More details are also presented in the Table 2. Thus, ICF algorithm has a significant improvement on the accuracy and quality of recommendation.

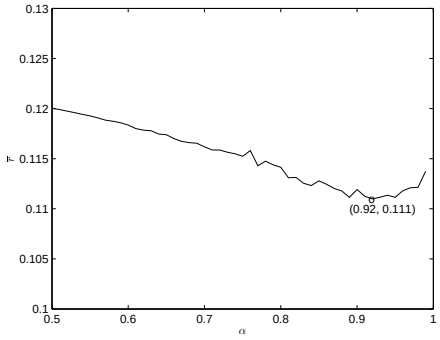
Table 1. Accuracy comparison of different algorithms.

MovieLens	$\overline{p}$	$\overline{Recall}$	$\overline{r}$	Novelty	Diversity
ICF	0.128	0.297	0.122	298	4.94
CF	0.115	0.247	0.132	327	7.48
GRM	0.049	0.126	0.217	389	/ *
COSCF	0.122	0.276	0.128	320	6.55
JaccardCF	0.129	0.297	0.122	308	5.56
Netflix	$\overline{p}$	$\overline{Recall}$	$\overline{r}$	Novelty	Diversity
ICF	0.102	0.338	0.069	1132	7.70
CF	0.093	0.311	0.074	1173	9.22
GRM	0.050	0.186	0.098	1373	/ *
COSCF	0.094	0.314	0.072	1172	9.15
JaccardCF	0.095	0.317	0.071	1172	9.13

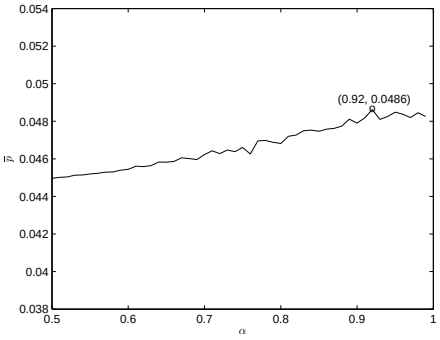
* this denotes that the evaluation metric is not available.

Table 2. Performance improvements of ICF algorithm compared to other algorithms.

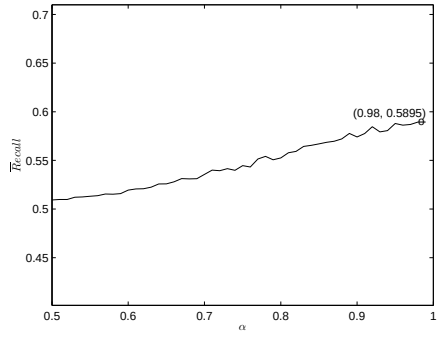
MovieLens	$\overline{p}$	$\overline{Recall}$	$\overline{r}$	Novelty	Diversity
CF	↑11.3%	↑20.4%	↑7.6%	↑8.9%	↑34.0%
GRM	↑161.2%	↑135.7%	↑43.8%	↑23.4%	/
COSCF	↑4.9%	↑7.6%	↑4.7%	↑6.9%	↑24.6%
JaccardCF	↓0.8%	↑ 0%	↑0%	↑3.2%	↑11.2%
Netflix	$\overline{p}$	$\overline{Recall}$	$\overline{r}$	Novelty	Diversity
CF	↑9.7%	↑8.7%	↑6.8%	↑3.5%	↑16.5%
GRM	↑104.0%	↑87.7%	↑29.6%	↑17.6%	/
COSCF	↑8.5%	↑7.6%	↑4.2%	↑3.4%	↑15.9%
JaccardCF	↑7.4%	↑6.6%	↑2.8%	↑3.4%	↑15.7%



(a) Average sorting accuracy



(b) Precision



(c) Recall

Figure 3. Performance comparison based on precision, recall and sorting accuracy with different α.

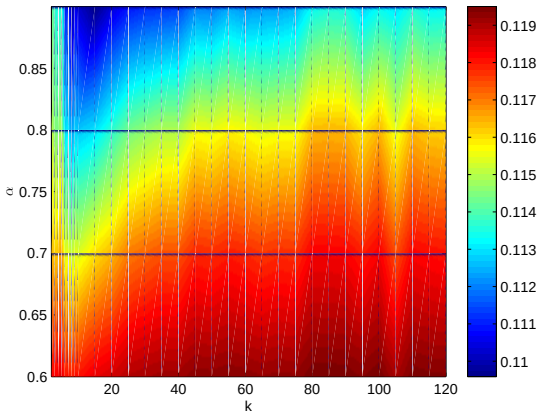


Figure 4. The plane chromatogram of (k, α) based on sorting accuracy.

5. Conclusions

In this paper, we propose an improved collaborative filtering personalized recommendation algorithm (ICF algorithm), based on k-means clustering and weighted similarity on reduced item space. Based on the traditional CF algorithm, the ICF algorithm establishes a weighted similarity model by reducing item space and clustering user information, which can extract the similarity information twice and get more user information. Then, the recommendation algorithm (ICF algorithm) is generated by the model that is optimized by PSO. Finally, a user-item recommendation list to achieve personalized recommendation is given, according to the ICF algorithm. Compared with the traditional CF algorithm, on the one hand, we solve the problem of data sparsity by reducing item space. On the other hand, we improve the degree of similarity recognition by secondary extraction of the similarity information between users, so that the ICF algorithm has a better recommendation effect. This conclusion is further verified by the experimental results on the benchmark data set, and the recommendation effect that the lCF algorithm is better than several algorithms (CF, GRM, COSCF and JaccardCF algorithm). However, the ICF algorithm is mainly used in the scoring system, and its application scope is limited. How to extend it to the general recommendation system is a topic that needs our further study.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (No.61563013), and the Natural Science Foundation of Guangxi (No. 2018GXNSFAA138095).

Conflict of interest

We declare that there are no conflicts of interest to this work.

References

1 Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, J. Wang, QoS ranking prediction for cloud services. *IEEE T. parall. distr.*, **24** (2012), 1213–1222. https://doi:10.1109/TPDS.2012.285

2 T. Liu, Z. He, A novel personalized recommendation algorithm by exploiting individual trust and item's similarities, *Appl. Intell.*, **52** (2022), 6007–6021. https://doi.org/10.1007/s10489-021-02655-1

3 L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, T. Zhou, Recommender systems, *Physics reports*, **519** (2012), 1–49. https://doi.org/10.1016/j.physrep.2012.02.006

4 Z. H. Deng, Z. H. Wang, J. Zhang, ROBIN: A novel personal recommendation model based on information propagation, *Expert syst. appl.*, **40** (2013), 5306–5313. https://doi.org/10.1016/j.eswa.2013.03.039

5 T, Zhou, J. Ren, M. Medo, Y. C. Zhang, Bipartite network projection and personal recommendation, *Phys. rev. E*, **76** (2007), 046115. https://doi.org/10.1103/PhysRevE.76.046115

6 R. Goyal, S. J. Goyal, Recommender system: An analytical report on decision making for large scale online social networks, *Materials*

Today: Proceedings, **47** (2021), 7145–7148. https://doi.org/10.1016/j.matpr.2021.06.311

7 I. Belkhadir, D. E. Omar, J. Boumhidi, An intelligent recommender system using social trust path for recommendations in web-based social networks, *Procedia computer science*, **148** (2019), 181–190. https://doi.org/10.1016/j.procs.2019.01.035

8 G. Linden, B. Smith, J. York, Amazon. com recommendations: Item-to-item collaborative filtering, *IEEE Internet comput.*, **7** (2003), 76–80. https://doi.org/10.1109/MIC.2003.1167344

9 S. Ding, C. Xia, C. Wang, D. Wu, Y. Zhang, Multi-objective optimization based ranking prediction for cloud service recommendation, *Decis. Support Syst.*, **101** (2017), 106–114. https://doi:10.1016/j.dss.2017.06.00

10 S. Yu, M. Yang, Q. Qu, Y. Shen, Contextual-boosted deep neural collaborative filtering model for interpretable recommendation, *Expert Syst. Appl.*, **136** (2019), 365–375. https://doi.org/10.1016/j.eswa.2019.06.051

11 T. Mohammadpour, A. M. Bidgoli, R. Enayatifar, H. H. Javadi, Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm, *Genomics*, **111** (2019), 1902–1912. https://doi.org/10.1016/j.ygeno.2019.01.001

12 T. M. Phuong, N. D. Phuong, Graph-based context-aware collaborative filtering, *Expert Syst. Appl.*, **126** (2019), 9–19. https://doi.org/10.1016/j.eswa.2019.02.015

13 C. Feng, J. Liang, P. Song, Z. Wang, A fusion collaborative filtering method for sparse data in recommender systems, *Inform. Sciences*, **521** (2020), 365–379. https://doi.org/10.1016/j.ins.2020.02.052

14 Y. Lv, Y. Zheng, F. Wei, C. Wang, C. Wang, AICF: Attention-based item collaborative filtering, *Adv. Eng. Inform.*, **44** (2020), 101090. https://doi.org/10.1016/j.aei.2020.101090

15 A. Gazdar, L. Hidri, A new similarity measure for collaborative filtering based recommender systems, *Knowledge-Based Systems*, **188** (2020), 105058. https://doi.org/10.1016/j.knosys.2019.105058

16 B. Alhijawi, G. Al-Naymat, N. Obeid, A. Awajan, Novel predictive model to improve the accuracy of collaborative filtering recommender systems, *Inform. Syst.*, **96** (2020), 101670. https://doi.org/10.1016/j.is.2020.101670

17 D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. Batista, M. N. García, A collaborative filtering method for music recommendation using playing coefficients for artists and users, *Expert Syst. Appl.*, **66** (2016), 234–244. https://doi.org/10.1016/j.eswa.2016.09.019

18 A. Hernando, J. Bobadilla, F. Ortega, A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model, *Knowledge-Based Systems*, **97** (2016), 188–202. https://doi.org/10.1016/j.knosys.2015.12.018

19 M. Singh, M. Mehrotra, Impact of biclustering on the performance of biclustering based collaborative filtering, *Expert Syst. Appl.*, **113** (2018), 443–456. https://doi.org/10.1016/j.eswa.2018.06.001

20 S. Kant, T. Mahara, V. K. Jain, D. K. Jain, A. K. Sangaiah, LeaderRank based k-means clustering initialization method for collaborative filtering, *Comput. Electr. Eng.*, **69** (2018), 598–609. https://doi.org/10.1016/j.compeleceng.2017.12.001

21 J. Chen, L. Wei, L. Zhang, Dynamic evolutionary clustering approach based on time weight and latent attributes for collaborative filtering recommendation, *Chaos, Solitons & Fractals*, **114** (2018), 8–18. https://doi.org/10.1016/j.chaos.2018.06.011

22 N. Vara, M. Mirzabeigi, H. Sotudeh, S. M. Fakhrahmad, Application of k-means clustering algorithm to improve effectiveness of the results recommended by journal recommender system, *Scientometrics*, **127** (2022), 3237–3252. https://doi.org/10.1007/s11192-022-04397-4

23 F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh, Recommendation systems: Principles, methods and evaluation, *Egypt. inform. j.*, **16** (2015), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

24 F. H. Del Olmo, E. Gaudioso, Evaluation of recommender systems: A new approach, *Expert Syst. Appl.*, **35** (2008), 790–804. https://doi.org/10.1016/j.eswa.2007.07.047

25 M. Nilashi, O. Ibrahim, N. Ithnin, Hybrid recommendation approaches for multi-criteria collaborative filtering, *Expert Syst. Appl.*, **41** (2014), 3879–3900. https://doi.org/10.1016/j.eswa.2013.12.023

26 R. Real, J. M. Vargas, The probabilistic basis of Jaccard's index of similarity. *Syst. biol.*, **45** (1996), 380–385. https://doi.org/10.1093/sysbio/45.3.380

AIMS Press