Hindawi
Journal of Applied Mathematics
Volume 2022, Article ID 2213173, 10 pages
https://doi.org/10.1155/2022/2213173

Research Article

Application of Improved K-Means Algorithm in Collaborative Recommendation System

Hui Jing

School of Intelligent Engineering, Nanjing City Vocational College, Nanjing 211200, China

Correspondence should be addressed to Hui Jing; jinghui@mjc-edu.cn

Received 24 August 2022; Revised 17 November 2022; Accepted 5 December 2022; Published 22 December 2022

Academic Editor: Fernando Simoes

With the explosive growth of information resources in the age of big data, mankind has gradually fallen into a serious "information overload" situation. In the face of massive data, collaborative filtering algorithm plays an important role in information filtering and information refinement. However, the recommendation quality and efficiency of collaborative filtering recommendation algorithms are low. The research combines the improved artificial bee colony algorithm with K-means algorithm and applies them to the recommendation system to form a collaborative filtering recommendation algorithm. The experimental results show that the MAE value of the new fitness function is 0.767 on average, which has good separation and compactness in clustering effect. It shows high search accuracy and speed. Compared with the original collaborative filtering algorithm, the average absolute error value of this algorithm is low, and the running time is only 50 s. It improves the recommendation quality and ensures the recommendation efficiency, providing a new research path for the improvement of collaborative filtering recommendation algorithm.

1. Introduction

With the continuous advancement of technology and the rapid development of the Internet, rich information resources are inundating everyone all the time, and both information producers and consumers are facing huge challenges [1]. Information producers need to deliver their information precisely to eligible target users, while as information consumers have to select effective data that satisfies themselves among the redundant and complicated mass of data, so the recommendation system appears. By analyzing users' historical data and actively providing them with news and products that meet their needs and interests, recommendation systems are able to filter information and provide users with personalized information services. It is playing an important role in various areas of social life such as current business systems [2]. Collaborative filtering recommendation algorithm is a comprehensive filtering algorithm, which takes the similarity of the item's attributes or user's ratings as the basis for personalized recommendation. It can handle unstructured complex objects without first extracting the content of the item. Collaborative filtering recommendation algorithms are capable of handling unstructured data and are gradually occupying a central position in recommendation systems. However, current collaborative filtering recommendation algorithms face a series of significant problems due to their own algorithmic characteristics, such as difficulties in ensuring real-time algorithm performance when dealing with huge data volumes. Ortega et al. developed a hybrid recommendation algorithm with multiclass classification algorithms and executed based on user rating behavior to improve the prediction and recommendation quality [3]. In addition, the artificial bee colony algorithm (ABC) has been effectively used in improving clustering performance due to its fewer parameters, simplicity, and ease of implementation and global merit seeking capability [4]. However, the K-means algorithm is prone to fall into local extremum and rely too much on the initial point selection, and the algorithm center determination time complexity is high. Therefore, in the second part of the study, the literature review of collaborative filtering algorithms at home and abroad is described. In the first section of the third part, the artificial bee colony algorithm is improved and combined with the K-means algorithm. The second section of

the third part is a collaborative filtering recommendation method based on the improved colony K-means algorithm. The fourth part is to verify the application effect of the proposed method. The fifth part is the conclusion of the study.

2. Related Work

In recent years, collaborative filtering recommendation algorithms and artificial bee colony K-means clustering models have received a great deal of attention among related professionals at home and abroad, and researchers have proposed many new methods for this purpose. Al-Bakri and Hassan proposed a modest approach to enhance data prediction by applying a user-based collaborative filtering algorithm to clustered data, and the results showed that the algorithm improved the recommendation system scalability [5]. Selvi and Sivasankar used supervised adaptive genetic networks to locate the most popular data points in clusters as a way to ensure simple and effective recommendations and reduce error rates. The effectiveness of the algorithm is demonstrated through experiments on the Netflix dataset [6]. Yang et al. proposed a time-weighted collaborative filtering algorithm with improved small-batch K-means clustering for sparse rating matrices and derived user scores with high recall and rating prediction accuracy to address shortcomings such as user interest bias in traditional collaborative filtering algorithms [7]. Wang et al. proposed a collaborative filtering algorithm incorporating temporal factors and applied it to score prediction and nearest neighbor selection with a time-weighted function. The results showed that the algorithm has good operational performance [8]. Najafabadi et al. performed neighbor selection for each user through user-based fuzzy clustering and a new similarity metric, and the results showed that it can improve the accuracy of recommendations to users [9]. Garanayak et al. developed a new recommendation system using K-means and item-based collaborative filtering techniques to filter out desired information segments based on people's preferences and concerns for the information [10].

Ashaduzzaman et al. proposed a clustering method that integrates multicriteria ratings into traditional recommender systems, rating in multidimensional situations such as auxiliary information, contextual information, and multiple criteria, and the results showed that the method was able to produce effective recommendations [11]. Ali and Wasid used the K-means algorithm for clustering and incorporated user-based rating criteria. The Mahalanobis distance metric was used to calculate the clustering similarity and generate a neighborhood set. Experiments showed that the algorithm improved the quality of recommendations [12]. Li et al. proposed a collaborative filtering algorithm based on category priority to address the problems of scalability and data sparsity of collaborative filtering recommendation algorithms and proposed a user-item priority ratio to calculate the priority ratio matrix. The recommendation accuracy of this algorithm was improved by 2.81% through experiments on the MovieLens dataset [13]. Onuean et al. set up recommendation items based on memory-based collaborative filtering techniques and used K-means clustering to cluster each data. The experimental results showed that the method has high accuracy in prediction and recommendation [14]. Zhu et al. proposed a fuzzy clustering-based method that evaluates the prediction-driven uncertainty and classifies based on existing data. Experimental results show that the method outperforms traditional collaborative filtering recommendation algorithms [15]. Chakraborty et al. addressed the problem of large bias in clustering results caused by initial guesses of clustering centers by combining the K-means algorithm with a volume metric algorithm and genetic arithmetic as a way to predict the optimal value of initial clustering centers. Experimental results showed that the algorithm improved prediction efficiency [16]. Daoudi et al. developed a new parallel K-means algorithm using a graphics processing unit to select the initial center of mass by using an open computing language in the programming environment and performing the initialization steps on the CPU in parallel. Experimental results showed that the method reduced the running time while maintaining quality [17].

In summary, the K-means algorithm has significant advantages in user clustering, and most researchers have also used K-means clustering by introducing it into traditional collaborative filtering recommendation algorithms to improve recommendation efficiency, but less research has been conducted on the combination of artificial bee colony algorithms with K-means. Therefore, the study will improve the swarm K-means model as a way to enhance the performance of recommendation systems.

3. Collaborative Filtering Recommendation Algorithm Based on Swarm K-Means Clustering Model

3.1. Improvement of Bee Colony Algorithm Based on K-Means Clustering Model. The artificial bee colony (ABC) algorithm is a colony intelligence optimization algorithm developed on the basis of simulating the foraging behavior of honey bee colonies. It has the advantages of simple operation, easy implementation, fast convergence, and few control parameters and is widely used in optimization problems such as function optimization and data mining [18]. The K-means algorithm, a distance-based hard clustering algorithm, is easy to end up with a local optimum and difficult to apply to data classification, despite its fast clustering speed and strong local search capability [19]. K-means objective criterion function is the distance from each point to the cluster cent. Solving for the extreme value of the function can iteratively adjusts the rules. K-means algorithm first selects k number of clusters and a dataset including n data, the initial cluster centers are selected randomly, the number of data samples is k, and the remaining nodes are $(n-k)$; then, they are divided into the class where the nearest centroid is located according to the distance of the remaining nodes to each initial cluster center. The new centroids of each cluster are obtained by calculating again, and it is judged whether the new cluster centroids change, i.e., whether the criterion function converges. If it converges, the algorithm ends; otherwise, it continues to the next iteration, as shown in Figure 1.

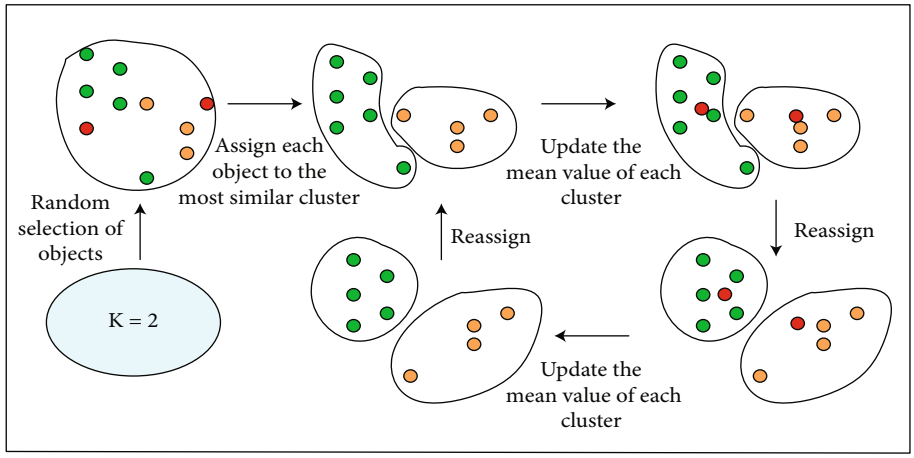


Figure 1: Schematic diagram of K-means.

The K-means algorithm divides the data samples mainly by measuring similarity, and the Euclidean distance formula for similarity between samples is given in

$$d(x_i, x_j) = \sqrt{\sum_{k-1}^{n} (x_{ik} - x_{jk})^2}. \tag{1}$$

In Equation (1), x_j and x_i are the data samples in the dataset, while x_{ik} and x_{jk} are the data samples in the k cluster. The objective criterion function of the K-means algorithm is the mean squared error, as shown in

$$E = \sum_{j=1}^{c} \sum_{k=1}^{n_j} \left\| x_k - m_j \right\|^2. \tag{2}$$

In Equation (2), x_k represents the data elements in the selected sample, E represents the sum of the mean squared deviations of the data elements, and m_j represents the cluster center of the j cluster. Cluster analysis is judged by the sum of squares of errors, as shown in Equation (3), with the aim of obtaining an optimal set of divisions that are as independent as possible between clusters and as compact as possible within clusters.

$$\begin{cases} J = \sum_{i=1}^{k} \sum_{\substack{j=1 \\ x_j \in C_i}}^{n} \operatorname{dis}(x_j, c_i)^2, \\ c_i = \dfrac{1}{N_i} \sum_{\substack{j=1 \\ x_j \in C_i}}^{n} x_j. \end{cases} \tag{3}$$

In Equation (3), x_j represents a data sample in the class C_i, c_i is the mean value of the data objects selected in the cluster C_i, N_i represents the number of data objects in the ith cluster, and $\operatorname{dis}(x_j, c_i)$ represents the Euclidean distance between c_i and x_j. The artificial bee colony algorithm is introduced. It is improved and combined with the K-means algorithm. The artificial bee colony algorithm is improved at three levels, fitness function, population initialization, and position update, and its conceptual correspondence with the K-means algorithm is shown in Table 1.

The initialization process of the artificial bee colony algorithm randomly generates nectar sources with twice the number of nectar sources already generated and randomly generates nectar source information within a given range of values. Finally, the fitness value of that source is calculated, as shown in

$$\begin{cases} X_{ij} = \beta_j + \operatorname{rand}(0, 1)\left(\alpha_j - \beta_j\right), \\ \text{fitness} = \begin{cases} \dfrac{1}{1 + \text{fitness}}, \\ 1 + \alpha\beta s(\text{fitness}). \end{cases} \end{cases} \tag{4}$$

In Equation (4), X is the nectar source information, fitness is the fitness value of the nectar source, rand $(0, 1)$ is a random number between $(0, 1)$, α_j represents the upper limit of the j dimensional data, and β_j is the lower limit of the j dimensional data. The conventional fitness function is shown in

$$\text{fitnessclassical}_I = \frac{\text{CN}_i}{J_i}. \tag{5}$$

In Equation (5), $\text{fitnessclassical}_I$ is the fitness value of the nectar source i, and CN_i represents the number of points in the i cluster. In order to combine the improved artificial bee colony algorithm with the K-means algorithm, it is necessary to construct a fitness function that makes K-means more efficient and faster clustering. Considering the intraclass distance and the number of points contained in each cluster as an influence, then the new fitness formula is given in

$$\text{fitnessnew}_I = \frac{\text{CN}_i}{J_i} + \frac{\text{CK}}{M}. \tag{6}$$

Table 1: Comparison of bee colony optimization and K-means.

K-means	Bee foraging behavior
Cluster center	Honey source location
Convergence speed of clustering algorithm	Search and foraging speed
Advantages and disadvantages of clustering center	Honey source richness
Cluster centers with high fitness	Rich food sources

In Equation (6), CK represents the sum of the distances from other cluster centers to the cluster center I, and M represents the number of cluster centroids. The hired bee performs a neighborhood search at its current location and selects a nectar source as shown in

$$\begin{cases} V_{ij} = X_{ij} + \theta_{ij}\left(X_{ij} - X_{kj}\right), \\ V_{ij} = \begin{cases} X_{ij} \text{ fitness}\left(V_{ij}\right) \le \text{fitness}\left(X_{ij}\right), \\ V_{ij} \text{ fitness}\left(V_{ij}\right) > \text{fitness}\left(X_{ij}\right). \end{cases} \end{cases} \tag{7}$$

In Equation (7), V represents the velocity, θ_{ij} is a random number between [-1, 1], and j is not equal to k. The ability of the swarm to obtain a higher quality honey source at a faster convergence rate depends on the location update formula. The existing artificial bee colony algorithm converges slowly in the later stages, so a global bootstrap factor is introduced and the new location update formula is shown in

$$V_{ij} = x_{ij} + r_{ij}\left(x_{lj} - x_{kj}\right) + \theta\left(x_{\text{best},j} - x_{ij}\right). \tag{8}$$

In Equation (8), v_{ij} represents a new location near x_{ij}. l, k, and j are random numbers obtained from a random formula, k and l are both equal to or not equal to i and mutually exclusive, $x_{\text{best},j}$ represents the current highest quality food source, and r_{ij} is a random number between [-1,1] and $\theta \in [0,1]$. The formula for calculating the probability when the full hired bee search is complete is given in

$$p_i = \frac{\text{fitnessnew}_I}{\sum_{i=1}^{N} \text{fitnessnew}_I}. \tag{9}$$

In Equation (9), p_i represents the probability, and fitness_I is the new fitness function value. The new location update formula shows that when the optimal location in the population is far from the location of the individual, the next iteration of the individual's search will increase the step size and approach the global optimal location at a faster rate, and vice versa, converge slowly. Commonly used external evaluation metrics for clustering are shown in

$$F_a = \frac{2P_aR_a}{P_a + R_a}. \tag{10}$$

In Equation (10), R_a is the completeness rate, P_a is the accuracy rate, andF_a is the external evaluation index of clusters. The weighted average of the F-measure of each category a gives F, as shown in

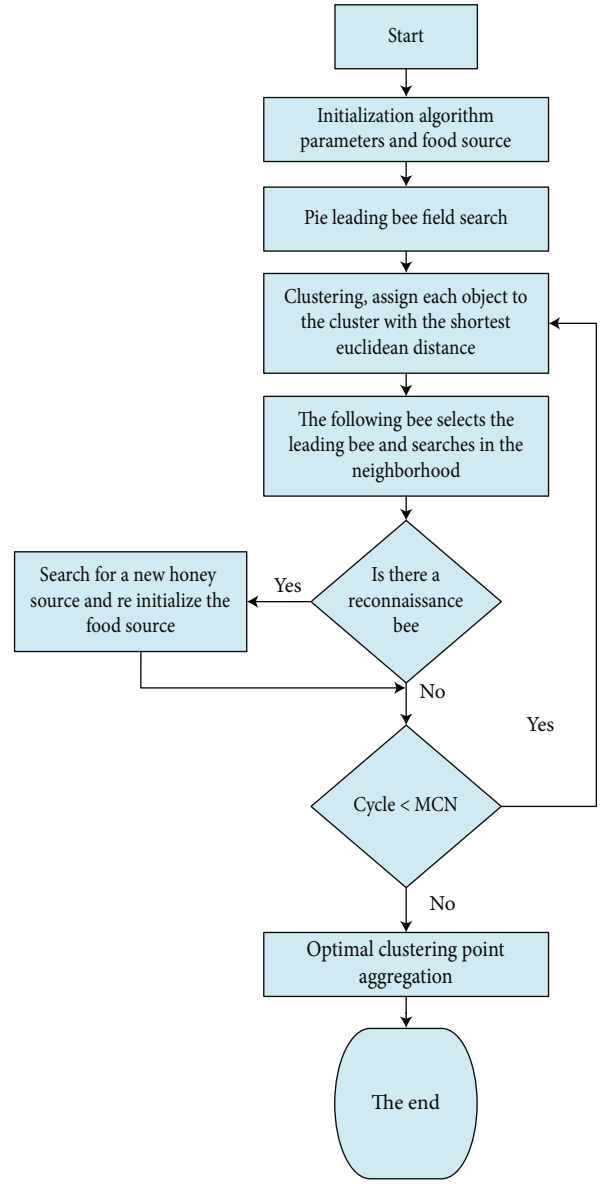


Figure 2: K-means clustering algorithm flow based on improved bee colony algorithm.

$$F = \frac{\sum_a [F_a \cdot |a|]}{\sum_a |a|}. \tag{11}$$

In Equation (11), $|a|$ represents the number of all objects in the classification$a$. So the flow chart of the improved algorithm is shown in Figure 2.

As can be seen from Figure 2, the improved algorithm has eight basic steps. Firstly, the number of scout bees, followers, and leaders is initialized, with equal numbers of followers and leaders; secondly, the initial colony is clustered, and the new fitness formula is used to derive fitness values, which are arranged in descending order. Thirdly, the leader

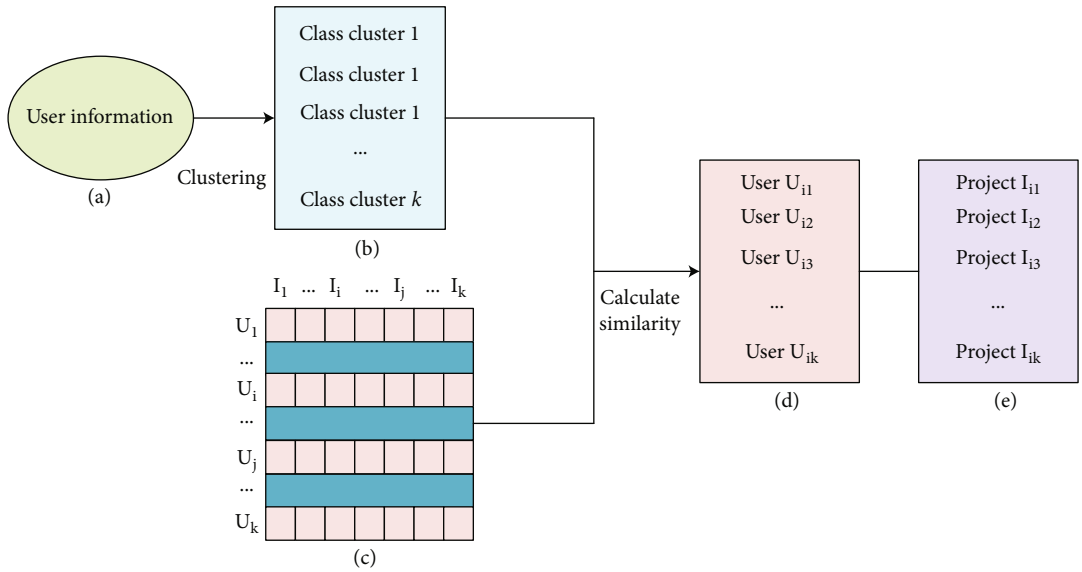


Figure 3: Schematic diagram of collaborative filtering recommendation based on bee colony K-means clustering.

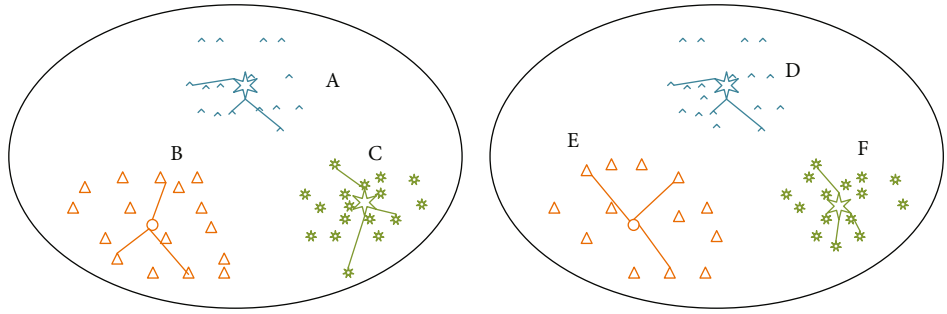
bees search the vicinity of the current location to obtain a new food source and then choose whether to keep the new food source based on the size of the fitness value compared to the original food source. Step 4 is to calculate the probability of following the bee P_i. Step 5 is to perform a nearest neighbor search after the following bee has selected the leader bee. Step 6 is to obtain the cluster center relative to the new food source after the nearest neighbor iteration and perform K-means clustering on the population. Step 7 is to check if there are any unrenewed food sources afterLimit iterations. Finally, depending on whether the number of iterations satisfies the termination condition, the algorithm is terminated algorithm or move to step 2.

3.2. Recommendation Algorithm Based on Improved Clustering Model. Collaborative filtering recommendation algorithm can filter information that cannot be automatically analyzed by the machine and has the ability to recommend new information. It has gradually become one of the most widely used and successful recommendation algorithms in the recommendation system [20]. However, collaborative filtering recommendation algorithms face the problems of data sparsity and cold start. The sparsity problem refers to the fact that users voluntarily give few reviews, and the common reviews of the same item by different users are even more scarce, so it tends to have low accuracy when calculating the similarity between items and between users. The cold start problem refers to the lack of historical evaluation data from users when new users and new items enter the recommender system or when a brand new system is just launched [21]. The problem of data sparsity is mainly compensated by filling in other useful information, which is a way to build effective models of user interests and item characteristics. Alternatively, the scoring data can be preprocessed by machine learning methods, such as matrix partitioning, matrix decomposition, and clustering, on the basis of existing data [22]. The cold-start problem then relies on the incorporation of trust relationships, background knowledge, and demographic information when calculating similarity, the integrated inclusion of item content information, and the proposal of new similarity metrics. The improved swarm K-means algorithm is therefore introduced into the collaborative filtering algorithm, where users are first clustered according to swarm K-means; i.e., the historical behavioral data of the initial user is analyzed and processed accordingly to find the set of users with similarities to the interest preferences of the user to be recommended. After clustering, clusters are obtained, and each cluster has a cluster center corresponding to it [23]. Then, the similarity between the target user and other users is calculated, and then, the recommendation list of the target user is formed based on the rating information of the nearest neighbor users. The principle of the algorithm is shown in Figure 3.

Table 2: Parameter setting.

Parameter	Value
Neighborhood mean parameter	10
Number of cluster centers	10
Control parameters limit	100
Maximum iteration times of clustering model	20

As can be seen from Figure 3, in the recommendation of user U, user clustering is performed based on the user's attribute information to obtain k clusters, and when user U exists in the class cluster j, the nearest neighbor set of T of U is obtained in the class cluster j based on the user-item rating matrix and calculated by similarity [24]. The similarity between users is mainly expressed by the cosine angle between user vectors, the magnitude of which is positively related to the degree of user similarity. The formula for



(a) Traditional fitness function (b) New fitness function

Figure 4: Clustering results of old and new fitness functions in the same data set.

calculating the similarity between the two is given in

$$\mathrm{sim}(u,v)=\cos\left(\vec{v}\cdot\vec{u}\right)=\frac{\vec{v}\cdot\vec{u}}{\left\|\vec{v}\right\|\cdot\left\|\vec{u}\right\|}=\frac{\sum_{i=1}^{n}R_{vi}\cdot R_{ui}}{\sqrt{\sum_{i=1}^{n}R_{vi}{}^{2}}\cdot\sqrt{\sum_{i=1}^{n}R_{ui}{}^{2}}}. \tag{12}$$

In Equation (12), both u and v represent the representative user, $\vec{u}$ represents the rating obtained by the user in the un dimensional space, $\vec{v}$ is the rating obtained by the user v in the n dimensional space, and sim represents similarity. Since the rating scales differ between users, the difference in rating scales is compensated by introducing an average user rating in the cosine similarity calculation, i.e., subtracting the average user score on the item. The modified cosine similarity formula is shown in

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in 1}(r_{vi}-\overline{r_v})\cdot(r_{ui}-\overline{r_u})}{\sqrt{\sum_{i\in 1}(r_{vi}-\overline{r_v})^{2}}\cdot\sqrt{\sum_{i\in 1}(r_{ui}-\overline{r_u})^{2}}}. \tag{13}$$

In Equation (13), r_{ui} represents the rating of user u on item i. $\overline{r_v}$ and $\overline{r_u}$ represent the average of the ratings of all items by user v and user u, respectively, i represents the elements in the item set, and $i\in I$. The items are then sorted in descending order of similarity, and a certain group of similar neighbors in the sort is selected together to form the set of nearest neighbors of the target user u. The predicted ratings of all items in the item set is shown in

$$P_{u,j}=\overline{r_u}+\frac{\sum_{v\in U_{ND}}\mathrm{sim}(v,u)\times\left(r_{vj}-\overline{r_v}\right)}{\sum_{v\in U_{ND}}\left|\mathrm{sim}(v,u)\right|}. \tag{14}$$

In Equation (14), U_{ND} represents the set of nearest neighbors, $P_{u,j}$ is the predicted score of user u on item j, and $\mathrm{sim}(u,v)$ is the similarity between user v and user u. The predicted scores are then used to obtain a score table, and recommendations are made based on the score table to obtain a recommendation list. The performance of the recommendation system is determined by the user's satisfaction level, which is also an indicator of the quality of the recommendation [25]. The most commonly used statistical accuracy measure is the mean absolute error (MAE), as shown in

Table 3: Comparison of MAE values between new and classical fitness functions.

Number of nearest neighbors	Fitnessnew	Fitnessclassical
10	0.845	0.825
15	0.818	0.815
20	0.775	0.833
25	0.745	0.803
30	0.693	0.800
35	0.730	0.750
Average	0.767	0.804
Winning times	4	2

Table 4: Characteristics of four test functions of improved bee colony algorithm.

Name	Search interval	Global minimum
Rosenbrock	[-100,100]	0
Griewank	[-600,600]	0
Rastrigin	[-100,100]	0
Sphere	[-100,100]	0

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|P_{ui}-Q_{ui}|}{N}. \tag{15}$$

In Equation (15), P_u is the predicted rating scale from user u, Q_u represents the actual rating scale, and N represents the number of test sets. The mean absolute error (MAE) is a measure of the difference between the true and predicted scores to measure the correctness. It is negatively correlated with the quality of the recommendation; i.e., a smaller MAE value means that the algorithm's recommendation quality is better, and conversely, a larger MAE value means that the algorithm's recommendation quality is worse.

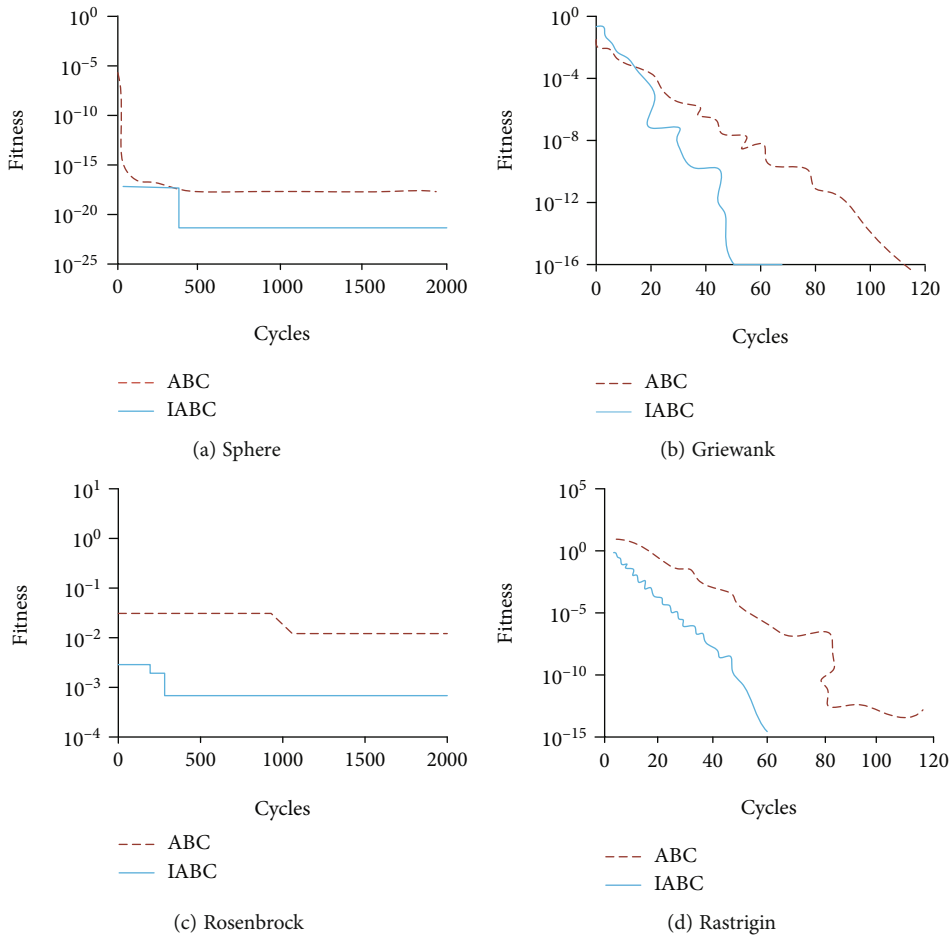


Figure 5: Change of fitness iteration trend before and after improved bee colony algorithm.

4. Application Effect Analysis of Collaborative Filtering Recommendation Algorithm Based on Bee Colony K-Means Clustering

An experimental analysis of the improved collaborative filtering recommendation algorithm was conducted. The configuration environment of the experiment includes Intel (R) Core (TM) i5-6200U CPU @ 2.30 GHz 2.40 GHz, 8.0 GB memory, 500G hard disk, Windows 10 64 bit operating system. Development platform is PyCharm platform based on Python 3.8.4. PyCharm is a powerful Python compiler. Its greatest advantage is that it combines multiple libraries (such as Matplotlib, pandas, and NumPy), which is simple and convenient. The experimental dataset was obtained from 90,000 ratings of 986 movies by 692 users on Douban, and the ratings were integers within [1, 5], with a positive correlation between the rating and the user's liking; i.e., a higher rating means that the user likes the movie more, while a lower rating means that the user likes the movie less. The specific parameters included in the experiment and the settings are shown in Table 2.

The improved collaborative filtering recommendation algorithm proposed in the study is based on the swarm K-means clustering model. The strength of the new fitness function directly affects the performance of the algorithm, so the traditional fitness function is compared with the new fitness function to test the performance of the improved algorithm. The clustering results of both under the same data set are shown in Figure 4.

Table 5: Parameter of improved ABC-K-means.

Name	Value
Colony size	20
Number of clusters in Iris	3
Maximum iterations	100
Number of clusters in Balance-scale	3
Number of clusters in Glass	6
Maximum mining times	10

In Figure 4, the larger asterisks represent the cluster centers, and points of the same shape are in the same class. From Figure 4(a), it can be seen that the distance of sample points within the class center is smaller, which achieves better intra-class compactness, but the separation between classes does not perform significantly, while the clustering results in Figure 4(b) not only have better compactness but also better separation, indicating that the overall performance of the

new fitness function is better than that of the traditional fitness function. The impact of both on the recommended results is shown in Table 3.

As can be seen from Table 3, the number of wins for the new fitness function and the traditional fitness function is 4 and 2, respectively, and the average MAE value forFitnessnew is 0.767 and 0.804 for Fitnessclassical, indicating that the new fitness function outperforms the traditional fitness function in terms of average value and number of wins. The results indicate that the new fitness function is able to improve the quality of recommendations. To further validate the performance of the improved swarm algorithm, four commonly used test functions, Griewank, Rastrigin, Sphere, and Rosenbrock, were tested. The characteristics of the four test functions are shown in Table 4.

In Table 4, Rastrigin and Griewank have similar characteristics, both being multipeaked functions, Sphere is a multipeaked convex function, and Rosenbrock is a convex function with consecutive single peaks. The global minimum 0 is obtained at$x_i = 0(i = 1, 2, \cdots, n)$ for all four. The iterative trends in fitness between the original swarm algorithm and the improved algorithm on Rastrigin, Griewank, Rosenbrock, and Sphere are shown in Figure 5.

From Figures 5(a) and 5(b), it can be seen that the iterative search for the optimal value of the original swarm algorithm has different degrees of slow convergence and local extremes in all four test functions; from Figures 5(d) and 5(c), it can be seen that the original artificial swarm algorithm requires a longer iteration time and more iterations to reach the same local optimal solution than the improved algorithm; and in Figure 5(d) and the fitness trend in Figure 5(b), the accuracy and precision of the original artificial bee colony algorithm in searching for the optimal solution is poor and differs from the improved bee colony algorithm in the optimal solution by multiple orders of magnitude. The initialization process of the improved algorithm is purposeful and introduces a global bootstrap factor, so its convergence speed and search accuracy are significantly higher than the original algorithm in the iterative search for the optimal solution. The improved swarm K-means algorithm was then tested for performance with the parameters set as shown in Table 5.

The number of datasets Iris, Balance-scale, and Glass was 150, 625, 625, and 214, respectively; the number of attributes was 4, 4, and 10, respectively; and the number of classifications was 3, 3, and 6, respectively. The convergence trends of the fitness values of the improved swarm K-means algorithm for 100 runs on the three datasets are shown in Figure 6.

As can be seen in Figure 6, on the Iris, Glass, and Balance-scale datasets, the magnitude of change in fitness of the improved swarm K-means algorithm is smaller over the course of the population iterations, and the new position update formula enables the algorithm to dynamically adjust the search step and gradually approximate towards the global optimum. The algorithm jumps out of the local optimum in 60 iterations on the Balance-scale dataset and reaches a position with a higher fitness value. Therefore,

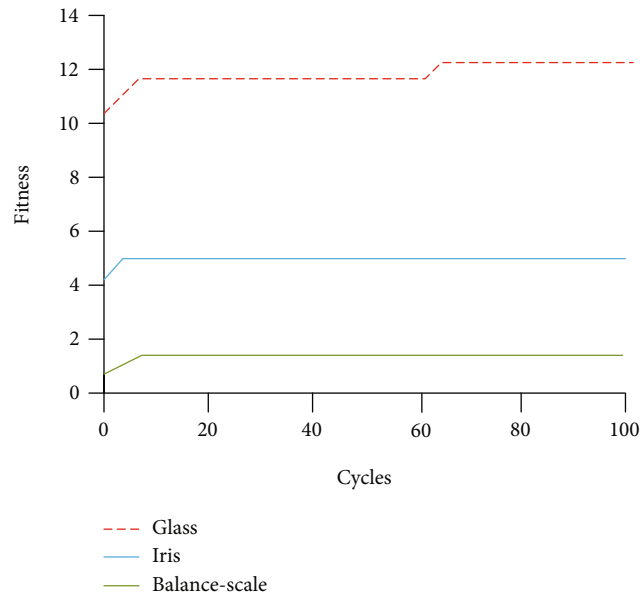


Figure 6: Convergence trend of fitness values of improved bee colony K-means algorithm on three data sets.

the algorithm is able to accurately obtain the global optimal solution in a shorter time with fewer iterations required, faster convergence, and higher stability for both the dataset Glass with a large attribute dimensionality and the dataset Balance-scale with a larger sample size. To verify the better recommendation quality of the collaborative filtering recommendation algorithm proposed in the study, it is compared with the user-based recommendation algorithm [26], the user-based clustering algorithm [27], and the ICCFRA algorithm [28]. Firstly, 560 users in the dataset were selected to form the training set and 321 to form the test set, and different numbers of nearest neighbors were set. The MAE results of the four recommendation algorithms are shown in Table 6.

In Table 6, "Number" represents the number of nearest neighbors. From Table 6, the MAE values of the algorithm proposed in the study are smaller than those of the user-based clustering algorithm and the user recommendation algorithm, and the MAE values of the ICCFRA algorithm gradually exceed the algorithm as the number of nearest neighbors increases. Therefore, the algorithm's recommendation results are more reliable, and the accuracy of the algorithm is more reliable when the amount of data increases. To verify the running efficiency of this algorithm, the running time of the algorithm was compared with the other three algorithms, and the running time of all four is shown in Figure 7.

As can be seen from Figure 7, the user-based recommendation algorithm takes significantly more time than the improved algorithm, with the highest at 150 seconds when the number of nearest neighbors is 30. The difference in running time between the collaborative filtering algorithm based on user clustering and the improved algorithm is smaller, with the highest time of the improved algorithm being 50 seconds. This is due to the fact that the improved algorithm

Table 6: Comparison of MAE results between the collaborative filtering recommendation algorithm based on bee colony K-means clustering and other three algorithms.

Number algorithm	User recommendation-based algorithm	User-based clustering algorithm	ICCFRA	Improved collaborative filtering recommendation algorithm
10	0.866	0.892	0.792	0.844
15	0.822	0.882	0.782	0.818
20	0.786	0.877	0.764	0.777
25	0.767	0.862	0.762	0.747
30	0.762	0.857	0.761	0.695
35	0.761	0.842	0.762	0.732

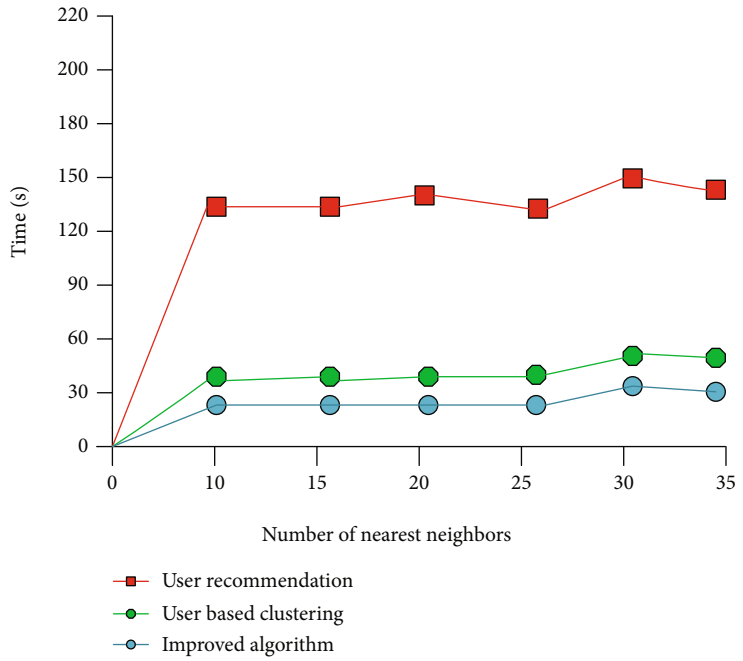


Figure 7: Running time comparison between collaborative filtering recommendation algorithm based on bee colony K-means clustering and other algorithms.

first clusters users and then builds user clusters, largely reducing the space for searching nearest neighbors. It improves the quality of recommendations as well as ensures operational efficiency.

5. Conclusion

Collaborative filtering recommendation algorithm is the most widely used and successful recommendation algorithm in the recommendation system, but its recommendation efficiency and quality are low at present. Therefore, an improved bee colony K-means clustering model is established and applied to the collaborative filtering recommendation algorithm to optimize the recommendation system. The experimental results show that under the same data set, the MAE value of the new fitness function is 0.767 on average, while that of the traditional fitness function is 0.804. In the four commonly used test functions Rosenbrock, Griewank, Rastrigin, and Sphere, the improved algorithm can obtain the same local optimal solution in a shorter iteration time and fewer iterations. In the iterative optimization process, the improved algorithm has higher convergence speed and search accuracy than the original algorithm. The MAE value of user clustering algorithm and user recommendation algorithm is larger than that of the improved algorithm, and the accuracy of the algorithm is more reliable when the number of nearest neighbors increases. In terms of running time, the improved algorithm has a maximum time of 50 seconds and has higher running efficiency. However, the user data included in the study is still relatively small, and more abundant information data is needed to determine the number of nearest neighbors.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

It is declared by the author that this article is free of conflict of interest.

References

[1] G. C. Krishnan, A. H. Nishan, and P. Theerthagiri, "K-means clustering based energy and trust management routing algorithm for mobile ad-hoc networks," *International Journal of Communication Systems*, vol. 35, no. 9, pp. 35–38, 2022.

[2] M. Yang, H. Mei, and D. Huang, "An effective detection of satellite images via K-means clustering on Hadoop system," *International Journal of Innovative Computing, Information & Control: IJICIC*, vol. 13, no. 3, pp. 1037–1046, 2017.

[3] F. Ortega, D. Rojo, P. Valdiviezo-Diaz, and L. Raya, "Hybrid collaborative filtering based on users rating behavior," *IEEE Access*, vol. 6, no. 99, pp. 69582–69591, 2018.

[4] N. Rahnema and F. S. Gharehchopogh, "An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering," *Multimedia Tools and Applications*, vol. 79, no. 43-44, pp. 32169–32194, 2020.

[5] N. F. Al-Bakri and S. Hassan, "Collaborative filtering recommendation model based on k-means clustering," *Al-Nahrain Journal for Engineering Sciences*, vol. 22, no. 1, pp. 74–79, 2019.

[6] C. Selvi and E. Sivasankar, "A novel adaptive genetic neural network (AGNN) model for recommender systems using modified k-means clustering approach," *Multimedia Tools & Applications*, vol. 78, no. 11, pp. 14303–14330, 2019.

[7] Y. Yang, Q. Liao, J. Wang, and Y. Wang, "Application of multi-objective particle swarm optimization based on short-term memory and K-means clustering in multi-modal multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 112, p. 104866, 2022.

[8] X. Wang, Z. Dai, H. Li, and J. Yang, "Research on hybrid collaborative filtering recommendation algorithm based on the time effect and sentiment analysis," *Complexity*, vol. 2021, no. 2, Article ID 6635202, 11 pages, 2021.

[9] M. K. Najafabadi, A. Mohamed, M. Nair, and S. M. Tabibian, "An effective collaborative user model using hybrid clustering recommendation methods," *Ingénierie des Systèmes D'Information*, vol. 26, no. 2, pp. 151–158, 2021.

[10] M. Garanayak, S. N. Mohanty, A. K. Jagadev, and S. Sahoo, "Recommender system using item based collaborative filtering (CF) and K-means," *International Journal of Knowledge-Based in Intelligent Engineering Systems*, vol. 23, no. 2, pp. 93–101, 2019.

[11] M. Ashaduzzaman, C. Jebarajakirthy, S. K. Weaven, H. I. Maseeh, M. Das, and R. Pentecost, "Predicting collaborative consumption behaviour: a meta-analytic path analysis on the theory of planned behaviour," *European Journal of Marketing*, vol. 56, no. 4, pp. 968–1013, 2022.

[12] R. Ali and M. Wasid, "Multi-criteria clustering-based recommendation using Mahalanobis distance," *International Journal of Reasoning-based Intelligent Systems*, vol. 12, no. 2, pp. 96–108, 2020.

[13] J. Li, K. Zhang, X. Yang et al., "Category preferred canopy-K-means based collaborative filtering algorithm," *Future Generation Computer Systems*, vol. 93, pp. 1046–1054, 2019.

[14] K. Onuean, S. Sodsee, and P. Meesad, "Top-k recommended items: applying clustering technique for recommendation," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 12, no. 2, pp. 106–117, 2019.

[15] J. Zhu, L. Han, Z. Gou, and X. Yuan, "A fuzzy clustering-based denoising model for evaluating uncertainty in collaborative filtering recommender systems," *Journal of the American Society for Information Science and Technology*, vol. 69, no. 9, pp. 1109–1121, 2018.

[16] S. Chakraborty, S. Raj, and S. Garg, "Selection of 'K' in K-means clustering using GA and VMA," *International Journal of Data Science*, vol. 4, no. 1, pp. 63–78, 2019.

[17] S. Daoudi, C. M. Anouar Zouaoui, M. C. El-Mezouar, and N. Taleb, "Parallelization of the K-means++ clustering algorithm," *Ingénierie des Systèmes D Information*, vol. 26, no. 1, pp. 59–66, 2021.

[18] I. H. Rifa, H. Pratiwi, and R. Respatiwulan, "Clustering of earthquake risk in Indonesia using k-medoids and k-means algorithms," *MEDIA STATISTIKA*, vol. 13, no. 2, pp. 194–205, 2020.

[19] W. Mohammed and A. Rashid, "Clustering approach for multidimensional recommender systems," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1122–1127, Singapore, 2018.

[20] A. Saklecha and J. Raikwal, "Enhanced K-means clustering algorithm using collaborative filtering approach," *Oriental Journal of Computer Science and Technology*, vol. 10, no. 2, pp. 474–479, 2017.

[21] M. Sridevi and R. R. Rao, "DECORS: a simple and efficient demographic collaborative recommender system for movie recommendation," *Advances in Computational Sciences & Technology*, vol. 10, no. 7, pp. 1969–1979, 2017.

[22] W. Mohammed and A. Rashid, "Fuzzy side information clustering-based framework for effective recommendations," *Computing & Informatics*, vol. 38, no. 3, pp. 597–620, 2019.

[23] M. Schoenfelder and E. Wammervold, "Disseminating and assessing implementation of the EULAR recommendations for patient education in inflammatory arthritis: a mixed-methods study with patients' perspectives," *RMD Open*, vol. 8, no. 1, pp. 954–962, 2022.

[24] S. Poudel and M. Bikdash, "Optimal dependence of performance and efficiency of collaborative filtering on random stratified subsampling," *Big Data Mining and Analytics*, vol. 5, no. 3, pp. 192–205, 2022.

[25] M. Verma and A. Rawal, "An enhanced item-based collaborative filtering approach for book recommender system design," *ECS Transactions*, vol. 107, no. 1, pp. 15439–15449, 2022.

[26] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," in *2018 9th international conference on information and communication systems (ICICS)*, pp. 102–106, Irbid, 2018.

[27] J. Cui, Z. Ding, P. Fan, and N. Al-Dhahir, "Unsupervised machine learning-based user clustering in millimeter-wave-NOMA systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7425–7440, 2018.

[28] X. J. Liu, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Computing*, vol. 20, no. 2, pp. 1281–1288, 2017.