Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7157075, 9 pages
https://doi.org/10.1155/2022/7157075

Research Article

Personalized Music Recommendation Algorithm Based on Spark Platform

Juan Sun

Department of Music, Handan University, Handan, Hebei 056005, China

Correspondence should be addressed to Juan Sun; sj03011015@hdc.edu.cn

Received 5 January 2022; Revised 19 January 2022; Accepted 26 January 2022; Published 17 February 2022

Academic Editor: Qiangyi Li

Copyright © 2022 Juan Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the shortcomings of traditional recommendation algorithms in dealing with large-scale music data, such as low accuracy and poor real-time performance, a personalized recommendation algorithm based on the Spark platform is proposed. The algorithm is based on the Spark platform. The K-means clustering model between users and music is constructed using an AFSA (artificial fish swarm algorithm) to optimize the initial centroids of K-means to improve the clustering effect. Based on the scoring relationship between users and users and users and music attributes, the collaborative filtering algorithm is applied to calculate the correlation between users to achieve accurate recommendations. Finally, the performance of the designed recommendation model is validated by deploying the recommendation model on the Spark platform using the Yahoo Music dataset and online music platform dataset. The experimental results show that the use of improved AFSA can complete the optimization of K-means clustering centroids with good clustering results; combined with the distributed fast computing capability of Spark platform with multiple nodes, the recommendation accuracy has better performance than traditional recommendation algorithms; especially when dealing with large-scale music data, the recommendation accuracy and real-time performance are higher, which meet the current demand of personalized music recommendation.

1. Introduction

With the continuous development of network and information technology, music works have exploded and gradually formed a vast music work library. Personalized recommendation has become a problem that needs to be solved by powerful music platforms [1, 2]. The recommendation service can filter the music data from the library according to the user's interest and behavior and recommend the corresponding music works intelligently under the premise of meeting the user's personalized needs. A sound recommendation system can save users' time and energy, improve user loyalty, and attract new users and make music platforms more competitive [3–5]. However, because there are many conditions to be met for music recommendation, which leads to the traditional single algorithm recommendation with low accuracy, poor real time, and colossal resource consumption, it cannot meet the current demand for a personalized offer of massive music data.

Based on this, some scholars have proposed improvements and optimization based on the original algorithm models such as collaborative filtering, content, and model and have achieved certain achievements. Lampropoulou et al. [6] proposed a music recommendation system based on SVM to accomplish the query of the same type of music resources through music content retrieval and collaborative filtering algorithm to achieve personalized music recommendation. However, the effect of recommendation capability is not satisfactory when dealing with large-scale dimensional data. Tian et al. [7] proposed a hybrid LX recommendation algorithm by integrating the logistic regression method and XG-Boost (eXtreme Gradient Boosting). It verified the effectiveness of the LX model by real music dataset, and experiments showed that the method has high accuracy. The experiments show that the method has high recommendation accuracy. However, due to the high spatial complexity of the XG-Boost preordering process, the implementation process is more complicated, and the

computational resources are more consumed. The real-time recommendation capability is not high. Using k-means clustering to find out users with similar attributes and pLSA (probabilistic Latent Semantic Analysis) technology to obtain the type of music users often listen to, the algorithm combines to achieve accurate recommendations with high efficiency [8] The accuracy and efficiency are high.

Although the above recommendation algorithm can accomplish music recommendations well, there is still a large room for improvement. Therefore, this paper proposes a K-means clustering music recommendation method based on AFSA optimization based on the Spark platform. The improved AFSA is used to solve the global optimization-seeking and local extremum problems of the K-means algorithm to improve the accuracy of multicategory clustering. The collaborative filtering algorithm is also used to deal with the similarity between users and improve the clustering effect and recommendation efficiency by the parallel computing capability of multiple nodes of the Spark platform.

2. Spark Platform Introduction

Apache Spark is a distributed computing engine specialized in processing large-scale data, with the advantages of fast data processing, strong stability, and scalability and support for multinode collaboration, which can complete data stream processing, interactive query, and batch processing in real time [9, 10]. As shown in Figure 1, Spark adopts the management model of master-slave nodes, and the nodes have the same computing capability in terms of functional structure. It is responsible for controlling the executor or driver process to complete resource management and task scheduling. The executor refers to the process of an application running on the work node.

Spark platform uses DAG (directed acyclic graph) job scheduling scheme. The dependencies between different data are specified by building a model of data relationships in RDDs (resilient distributed datasets). To facilitate the iterative computation of data, the results of the distributed analysis are usually stored in memory. The RDD is a read-only, partitioned, and immutable dataset. Since Spark uses the RDD computation model to do most of the data computation, the model calls more methods and does not need to focus on the underlying scheduling details [11]. This fundamentally avoids the problem of low computing efficiency caused by multiple iterations in the clustering process, dramatically improves the effect of data clustering and recommendation, and meets the demand for real-time and intelligent music recommendations.

3. K-Means Clustering

K-means clustering is one of the commonly used algorithms in data mining. The algorithm uses the distance between data sample points to measure similarity. The closer the distance, the higher the similarity of the data. Set up a dataset $C = \{c_i, i = 1, 2, \ldots, m\}$ with m sample points; each sample has an N data dimension feature description [12]. The K-means algorithm aims to allocate all the data samples in the n dataset and divide the pieces into corresponding K clusters according to the principle of similarity, cluster $D = \{d_k, k = 1, 2, \ldots, K\}$. The error of clustering is evaluated using the sum of squared errors, which is computed repeatedly until the sum of squared errors of clustering is the smallest [13]. The algorithm flow is shown in Figure 2.

There are more methods to calculate the distance between data sample points, and the K-means algorithm mainly uses the Euclidean distance to measure the similarity between data samples. The Euclidean distance between two data samples in dataset C can be expressed as [14]

$$\operatorname{dist}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{1}$$

The mean of the N data samples within cluster d_k is calculated and considered as the center of mass h_k of the cluster, and the expression is

$$h_k = \frac{\sum_{c_i \in d_k} c_i}{N}. \tag{2}$$

The criterion for clustering sum of the squared errors G as data samples:

$$G = \sum_{k=1}^{K} \sum_{c_i \in d_k} |c_i - h_k|. \tag{3}$$

The implementation of the K-means algorithm is relatively simple. Still, the number of clusters depends on manual experience setting, which takes a long time when dealing with large-scale data such as music and video, and the accuracy of clustering is not ideal. Moreover, the initial clustering centroid of the algorithm is selected randomly, which is prone to the situation that the clustering results vary greatly. If a relatively isolated point is selected as the initial clustering centroid, it will directly affect the accuracy and efficiency of clustering [15, 16]. Considering the actual application requirements, the K-means algorithm needs to be improved to enhance the clustering accuracy and efficiency.

4. Improved Artificial Fish Swarm Algorithm

4.1. Artificial Fish Swarm Algorithm. AFSA is a bionic optimization algorithm proposed by studying the intelligent behavior of fish. Based on the environmental stimulus information received by the fish, an artificial fish is constructed, and its feeding, gathering, and tailing behaviors are simulated to accomplish automatic optimization of the solution space [17, 18]. In this paper, AFSA is used to select the initial cluster center of mass for K-means clustering. The data sample of the desired cluster is used as the artificial fish. When the fish in the group find food, the other individuals in the collection adjust their positions according to their state and food position and finally obtain the optimal global value, which is the initial center point of the cluster. The principle of artificial fish seeking is shown in Figure 3.

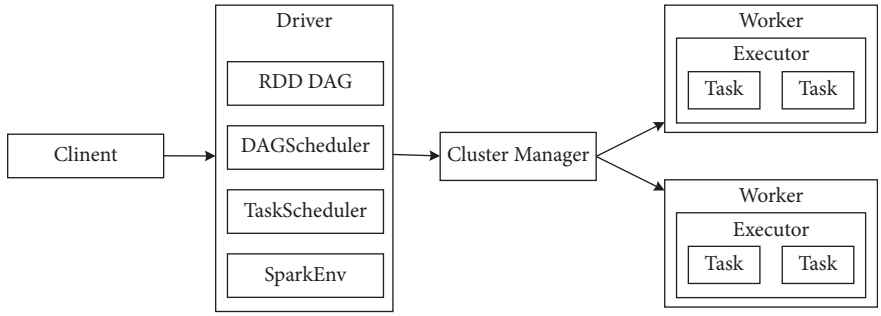

Figure 1: Traditional character recognition process.

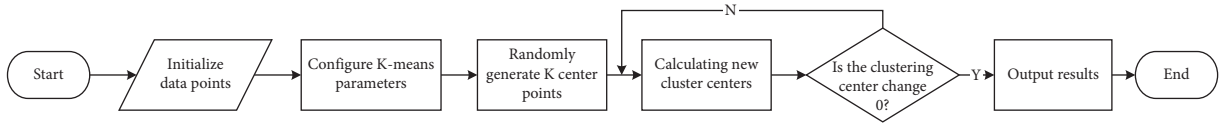

Figure 2: Flow of K-means algorithm.

Let $Y_i(t)$ be the spatial position of the artificial fish i at the t moment; $\Delta Y_i(t+1)$ denotes the change in spatial part of the fish i at the $t+1$ moment compared to the t moment; then, the random movement behavior of the artificial fish Y_i can be expressed as

$$\Delta Y_i(t+1) = \text{Rand} * \text{Step} * \left[Y_i(t+1) - Y_i(t)\right], \tag{4}$$

where Rand() is a random function and *Step* denotes the update of the artificial fish stocks. Then, a specific state Y_j within the visual of the artificial fish Y_i is randomly selected as

$$Y_j = Y_i + \text{Visuanl} * \text{Rand}. \tag{5}$$

Calculate the objective function values Z_i and Z_j for Y_i and Y_j. If Z_j is better than Z_i, then Y_i is one step ahead in the direction of Y_j. There are many kinds of filtering algorithms:

$$Y_i^{t+1} = Y_i^t + \frac{Y_j - Y_i^t}{\left\|Y_j - Y_i^t\right\|} * \text{Step} * \text{Rand}. \tag{6}$$

If Z_j is worse than Z_i, fish Y_i reselects another state Y_j in its field of view and determines whether it can advance. After repeated attempts of Tray-number times, if it still cannot get the condition to satisfy the advancement, the random behavior is executed.

Let the food concentration function in spatial coordinates be $f(y)$, the coordinate food concentration be $f_{\max}$, and the food concentration at the Y_i coordinate of the artificial fish be $f(Y_i)$. Based on the maximum value of the food concentration $f(y)$ within the field of view of the fish, determine whether the food location Y_o is in the visible range of the fish, and if it is visible, perform the trailing behavior:

$$Y_i^{t+1} = Y_i^t + \frac{Y_O - Y_i^t}{\left\|Y_O - Y_i^t\right\|} * \text{Step} * \text{Rand}. \tag{7}$$

4.2. Optimization of Artificial Fish Swarm Algorithm

4.2.1. Optimization of Fish School Behavior. In the foraging behavior of AFSA, Y_j is any state within the current field of view of the artificial fish. The artificial fish performs random behavior when the selected form cannot satisfy the forward condition. The existence of unexpected behaviors in the algorithm will result in lower accuracy of the foraging results; the convergence process of the algorithm in the later stage is prone to the phenomenon of artificial fish searching in a roundabout way at the global extremum, which leads to invalid calculation. At the same time, when the artificial fish fails in clustering and tail-chasing behavior, it performs foraging behavior, and the artificial fish has more invalid forward behavior, which makes the convergence time of the algorithm longer and the iterative computation more significant. Therefore, this paper optimizes the fish swarming behavior in the original algorithm in the following two aspects to further improve the algorithm's performance.

(1) When the foraging behavior fails, the artificial fish Y_i moves one step forward in the direction of the relatively better state in the bulletin board record. The optimized expression for the foraging behavior is

$$Y_i^{t+1} = Y_i^t + \left[Y_{\text{better}}^{t+1} - Y_i^t\right] * \text{Step} * \text{Rand}, \tag{8}$$

where Y_{better} is the relatively better state in the bulletin board record and $[Y_{\text{better}}^{t+1} - Y_i^t]$ is the increment of that state.

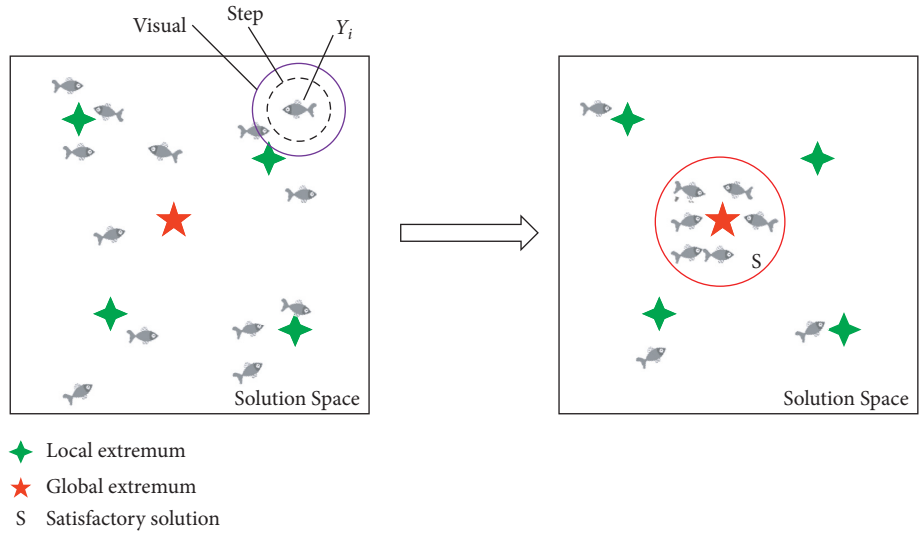

Figure 3: Principle diagram of artificial fish seeking advantage.

The optimized foraging behavior discards the phenomenon of artificial fish randomly advancing and provides a better possibility of running for fish movement, which can effectively avoid artificial fish falling into local optimum and can further improve the accuracy of the search results.

(2) When the clustering behavior or tail-chasing behavior fails, the artificial fish no longer performs the foraging operation but executes the random swimming behavior, and the optimized unexpected moving behavior is defined and expressed as

$$Y_i^{t+1} = Y_i^t + \text{Step}(t+1). \tag{9}$$

By optimizing the forward rule after the failure of clustering and tailing behavior, we can reduce the invalid brash behavior of artificial fish, thus effectively reducing the algorithm running time and fundamentally reducing the computational complexity.

4.2.2. Adaptive Field of View and Step Size. The two parameters of the artificial fish's field of view and step length determine the current search range and the subsequent moving speed. When artificial fish are searching far away from the extreme point, they should have a larger field of view and step to improve search speed. Also, when the fish in the area nearer to the extreme value point are searched, the values of the field of view and step length should be small to ensure the accuracy of the search. Considering the global fixed values of the field of view and step length in the original algorithm, it is easy to fall into the dire situation of blind search [19, 20]. In this paper, the adaptive value of both is accomplished by introducing the decay factor.

$$\sigma = \frac{Y_j - Y_i}{\left\| Y_j - Y_i \right\|} * \text{Rand}, \tag{10}$$

where σ is the attenuation factor with forwarding direction to constrain the step length, the field of view, and forward movement of the artificial fish, $\sigma \in [0, 1]$. Then, the equations of adaptive step length and adaptive decay field of view can be expressed as

$$\begin{aligned} \text{Step}(t+1) &= \text{Step}(t) * \sigma, \\ \text{Visual}(t+1) &= \text{Visual}(t) * \sigma. \end{aligned} \tag{11}$$

The iterative process of AFSA is a continuous process of artificial fish aggregation to the global extremum. By introducing a decay factor, the fish can dynamically adjust the field of view and step size according to the external environmental information during the search process. In the early stage of the algorithm's operation, the artificial fish use a larger field of view and step size to conduct a coarse search in a wide range near the optimal value, thus speeding up the individual search speed and making the algorithm converge quickly. After several iterations, the artificial fish gradually approaches the extreme value. At the end of the iteration, the field of view and step size is adjusted adaptively by the attenuation factor to enhance the local search ability and reduce the oscillation range. The artificial fish can quickly find extreme global points with high accuracy. Especially in the case of multidimensional data, the algorithm of finding the optimal effect with the introduction of the decay factor is more prominent.

4.3. User-Based Collaborative Filtering. After clustering by the AFSA-optimized K-means algorithm, the user's rating matrix of all resources and services to be recommended can be obtained. The collaborative filtering algorithm can find users with the same rating items in this rating matrix and calculate the similarity to complete practical personalized music recommendations. There are many kinds of filtering algorithms, but PPMCC (Pearson product-moment correlation coefficient) solves the difference of each dimension, and the similarity calculation process focuses on the whole so that it can complete the multidimensional examination of the same user [21]. So, this

paper uses PPMCC to calculate the similarity between users. The expression is

$$\mathrm{sim}(u_u, u_v) = \frac{\sum_{i_c \in I(u,v)} (r_{uc} - \overline{r}_u)(r_{vc} - \overline{r}_v)}{\sqrt{\sum_{i_c \in I(u,v)} (r_{uc} - \overline{r}_u)^2} \times \sqrt{\sum_{i_c \in I(u,v)} (r_{vc} - \overline{r}_v)^2}}, \quad (12)$$

where $I(u, v)$ represents the set of shared items jointly rated by users u_u and u_v, r_{uc} and r_{vc} represent the ratings of items i_c by users u_u and uv, respectively, and $\overline{r}_u$ and $\overline{r}_v$ represent the average ratings of all calculated items by users u_u and uv, respectively.

Table 1 shows the user-user similarity matrix W calculated by the PPMCC.

4.4. Music Recommendation Process Based on Spark. The recommendation process of K-means clustering mining based on AFSA optimization is shown in Figure 4. First, the requirement analysis is performed for the music personalization recommendation task. The Spark platform deploys distributed nodes of a suitable size to build the K-means clustering operation model. The AFSA algorithm is applied to optimize the centroids of the initial K-means clustering and obtain the clustering results. According to the scoring model between users and users and users and music attributes, the collaborative filtering algorithm calculates the similarity between users. In turn, the personalized recommendation of music resources is completed.

5. Experiment

5.1. Experimental Dataset and Clustering Center K Values. To verify the performance of the AFSA-optimized K-means clustering personalized music recommendation algorithm on the Spark platform, public and proprietary datasets are tested separately. The public dataset is Yahoo Music, a classical dataset for testing recommendation systems which can well validate the recommendation performance of clustering mining. The dataset contains large-scale user and music resource data to validate the recommendation system's performance adequately. The experimental dataset is shown in Table 2. The Spark platform contains one master node and nine work nodes with the same hardware configuration, CPU: I5-11400: memory: Kingston 3600 8gb ∗ 2; and system: IOP 4.2 Version, including Spark Version 1.61.

The clustering center K value is more sensitive to the music scoring matrix and has a more significant impact on the stability of collaborative filtering recommendations. Therefore, this paper differentiates the K values and uses RMSE (root mean squared error) as an evaluation index to verify the recommendation accuracy under different K values [22]. The expression of RMSE is

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{b=1}^{N} (p_b - q_b)^2}{N}}, \quad (13)$$

where p_b denotes the predicted score, q_b denotes the actual score, and N represents the number of items. Smaller RMSE values indicate better algorithm performance.

The data from the Yahoo Music dataset were randomly taken and formed into three sample sets of different sizes YM-Data1 (13.82 MB), YM-Data2 (43.39 MB), and YM-Data3 (140.24 MB), and then experiments were conducted with the three sample data, respectively, to derive different K-value recommendation accuracy RMSE values, as shown in Figure 5.

As shown in Figure 5, the RMSE value decreases and increases as the K value increases, and the RMSE value is optimal for the three sample sets of YM-Data1, YM-Data2, and YM-Data3 when the K value is 20, 20, and 22. When increasing the K values, the RMSE values gradually become more extensive, and the recommendation stability worsens. Therefore, in the subsequent training process, the range of K values is set to [20, 22].

5.2. Performance of the Recommendation Algorithm

5.2.1. Public Datasets. The K value is set to 20, and the designed A-optimized K-means algorithm is used for clustering mining. The recommendation algorithm model is used for training the Yahoo Music sample set. All master nodes and work nodes are involved in the computation during training. The recommended performance of the three sample sets is shown in Table 3.

As shown in Table 3, the difference in sample set size has little influence on the recommended usage time and accuracy, and it has good real-time recommendation ability because the data are loaded into the distributed memory of the Spark cluster host and the conversion iteration completes faster, especially when the sample data size is small.

5.2.2. Online Music Dataset. To further validate the recommended performance of this method, a dataset from an online music platform is used for testing. First, four sets of sample sets with different capacities based on music platform are built: KW-Data1 (10.75 MB), KW-Data2 (474.84 MB), KW-Data3 (1.62 GB), and KW-Data4 (6.65 GB). Then, cluster mining is applied to each data sample to get the score between all users and music resource attributes. Finally, the recommended results are obtained by using the collaborative filtering algorithm. KW-Data2 is used as an example to visualize its clustering results to facilitate the observation of the clustering effect, and the result is shown in Figure 6.

To make the results comparable, the algorithm performance was experimented with differentially setting the number of clustering centers for K taking values of 18, 20, and 22. The obtained results are shown in Table 4.

As can be seen from Table 3, the accuracy rate stays above 90% when K values are selected from the optimization algorithm, and the recommendation accuracy rate improves by 20.3%–21.7% compared with the randomly selected K values. Since the Spark platform uses multinode computing, the recommendation time gradually increases with the increase of sample capacity under different K values. Still, there is no rapid growth phenomenon, in line with the actual rule.

Table 1: User-user similarity matrix W.

U	U			
	u_1	u_2	...	u_m
u_1	sim (u_1, u_1)	sim (u_1, u_2)	...	sim (u_1, u_m)
u_2	sim (u_2, u_1)	sim (u_2, u_2)	...	sim (u_2, u_m)
...	...	...	...	...
u_m	sim (u_m, u_1)	sim (u_m, u_2)	...	sim (u_m, u_m)

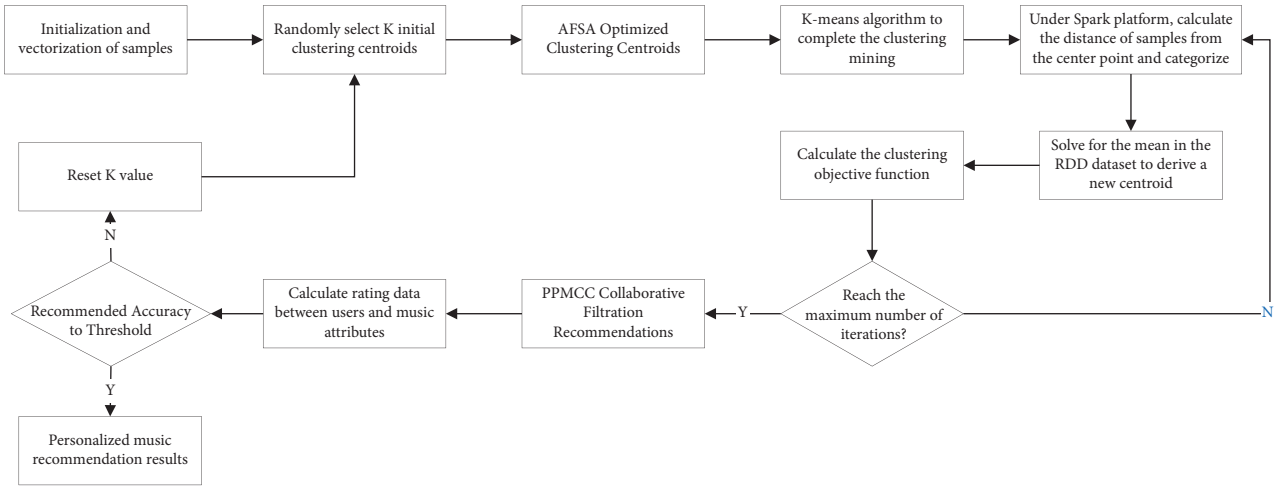

Figure 4: Personalized music recommendation algorithm flow based on Spark platform.

Table 2: Information on the datasets used for the experiments.

Dataset	Number of users	Number of works	Number of ratings	Rating density	Rating level
Yahoo Music	7400	3000	15335	0.0006	1, 2, ..., 100
Online music platform	11406	34668	170321	0.0152	1, 2, ..., 10

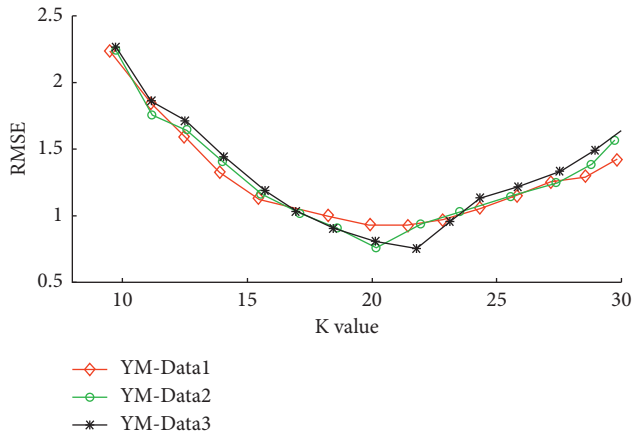

Figure 5: Recommended accuracy RMSE values for different K values.

Table 3: Recommendation performance based on Yahoo Music dataset.

Sample set	Accuracy (%)			Recommended time (s)
	Minimum value	Average value	Maximum value	
YM-Data1	82.872	85.097	87.233	0.973
YM-Data2	82.165	85.714	87.792	0.944
YM-Data3	83.281	85.639	87.935	0.984

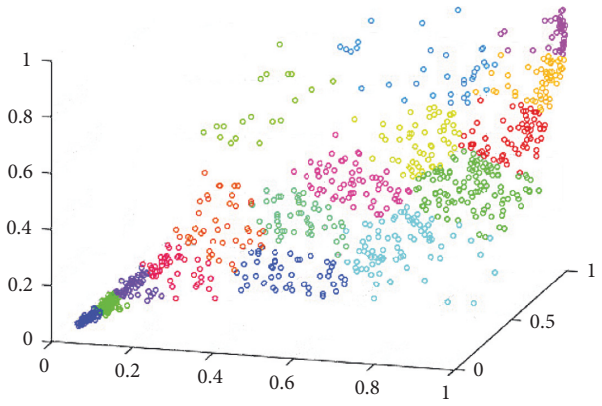

Figure 6: Visualization of clustering results of KW-Data3.

Table 4: Recommended performance for different K values.

Sample set	Capacity	$K = 18$		$K = 20$		$K = 22$	
		Accuracy (%)	Recommended time (s)	Accuracy (%)	Recommended time (s)	Accuracy (%)	Recommended time (s)
KW-Data1	10.75 MB	84.407	3.375	93.524	1.273	93.177	1.451
KW-Data2	474.84 MB	81.762	7.243	93.007	3.104	93.090	3.329
KW-Data3	1.62 GB	80.155	9.716	91.808	4.774	91.832	4.808
KW-Data4	6.65 GB	70.094	19.458	90.450	8.635	90.892	8.924

5.3. Spark Platform Acceleration Performance Testing. The higher the speedup ratio is, the more parallelized the platform is. The acceleration ratio expression is

$$\text{Speed up}(\rho) = \frac{T_1}{T_\rho}, \tag{14}$$

where T_1 denotes the recommended time for a single node and T_ρ denotes the time used for multinode parallel computation under the Spark platform. The speedup performance at the different number of nodes is shown in Table 5.

From Table 5, we can see that the more the number of worker nodes is, the more significant the speedup effect is due to the excellent parallelization capability of Spark platform, and the larger the volume of the sample set, the more influential the impact of the number of nodes on the speedup ratio.

5.4. Performance Comparison of Different Recommendation Algorithms. Five hundred samples were randomly selected from each online music platform sample set. A new selection set KW-Data5 was constructed to test the SVM classification algorithm proposed in [6], the XG-Boost algorithm presented in [7], and the algorithm in this paper to ensure the comparability of the algorithms and further verify the performance of the optimized K-means clustering based on the improved artificial fish swarm algorithm. The recommendation performance of the three algorithms is shown in Figure 7.

As can be seen from Figure 7, the recommendation accuracy of each algorithm increases with the number of iterations and then gradually stabilizes. However, the recommendation accuracy of this method reaches 96.23%, which is 6.77% and 15.32% higher than that of XG-Boost and SVM, respectively, and the recommendation performance is more excellent. The convergence speed of different algorithms: the method in this paper tends to be stable after 245 iterations; the SVM tends to be stable after 230 iterations; XG-Boost algorithm completes convergence after 295 iterations, which is slower. Therefore, based on the comprehensive recommendation performance comparison results, it can be concluded that the improved AFSA-based optimized K-means clustering method has good recommendation ability, can better balance the relationship between accuracy and convergence speed, and is suitable for handling large-scale personalized music recommendation work.

Table 5: Comparison of acceleration performance with different number of nodes.

Sample set	Capacity	Number of work nodes (pcs)	Speedup	Sample set	Capacity	Number of work nodes (pcs)	Speedup
		1	1.000			1	1.000
KW-Data1	10.75 MB	5	1.002	KW-Data2	474.84 MB	5	1.007
		10	1.003			10	1.015
		1	1.000			1	1.000
KW-Data3	1.62 GB	5	5.486	KW-Data4	6.65 GB	5	19.775
		10	12.837			10	48.093

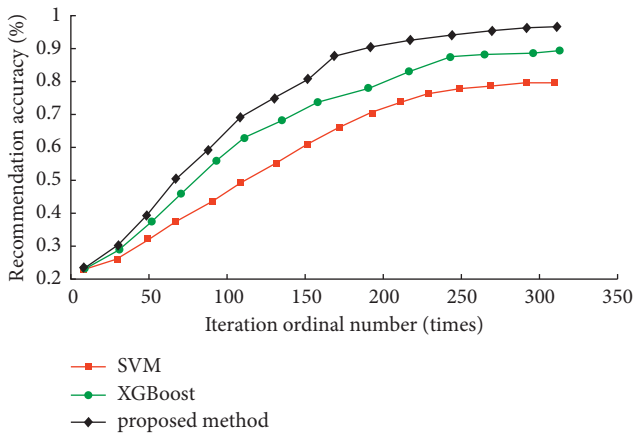

Figure 7: Recommendation performance of different algorithms.

6. Conclusion

To address some shortcomings of the traditional music recommendation algorithm, the improved AFSA is applied to solve the global optimization and local extremum problems of the K-means algorithm, thus improving the effect of multicategory clustering, completing the personalized recommendation of music by collaborative filtering algorithm, and improving the speed of large-scale data processing by using the parallelized computing feature of Spark distributed platform. The experimental results show that the AFSA-optimized K-Means clustering mining algorithm has good recommendation capability, and the recommendation accuracy is improved by 6.77% and 15.32% compared with the commonly used SVM and XG-Boost methods; the algorithm iterates faster to meet the requirements of real-time recommendation, which is enough to improve the recommendation efficiency of personalized music effectively and can be used as a reference for music, video, and similar fields, with good guiding significance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] H. Y. Hung-Yi Lo, J. C. Ju-Chiang Wang, H. M. Hsin-Min Wang, and S. D. Shou-De Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 518–529, 2011.

[2] J. Bae and J. Kim, "Deep learning music genre automatic classification voting system using softmax," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 1, pp. 27–32, 2019.

[3] W. A. Yudiana, M. Ariyanti, and A. Alamsyah, "Wisdom of the crowd" as personalized music recommendation model for langit musik service,"vol. 1, pp. 345–350, in *Proceedings of the 2019 International Conference on Information Management and Technology (ICIMTech)*, vol. 1, pp. 345–350, IEEE, Denpasar, Indonesia, 2019 August.

[4] D. Xu, Y. Dai, and W. Dai, "Smart recommendation of personalized cloud music service based on mental model," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, Chengdu, China, 2019 May.

[5] J. Geng, "Personalized analysis and recommendation of aesthetic evaluation index of dance music based on intelligent algorithm," *Complexity*, vol. 2021, Article ID 1026341, 15 pages, 2021.

[6] P. S. Lampropoulou, A. S. Lampropoulos, and G. A. Tsihrintzis, "A mobile music recommender system based on a two-level genre-rating SVM classifier enhanced by collaborative filtering," *New Directions in Intelligent Interactive Multimedia Systems and Services - 2*, Springer, Berlin, Heidelberg, pp. 361–368, 2009.

[7] H. Tian, H. Cai, J. Wen, S. Li, and Y. Li, "A music recommendation system based on logistic regression and eXtreme gradient boosting," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, Budapest, Hungary, 2019, July.

[8] L. He and K. He, "Efficient memory-bounded optimal detection for GSM-MIMO Systems," to appear," *IEEE Transaction Communication*, vol. 99, pp. 1–15, 2022.

[9] F. Yang, H. Wang, and J. Fu, "Improvement of recommendation algorithm based on collaborative deep learning and its parallelization on Spark," *Journal of Parallel and Distributed Computing*, vol. 148, pp. 58–68, 2021.

[10] P. Singh, K. Dutta, R. Kaye, and S. Garg, "Music listening history dataset curation and distributed music recommendation engines using collaborative filtering," in *Proceedings of the ICETIT 2019*, pp. 623–632, Springer, Delhi, India, September 2020.

[11] X. Lai and L. Fan, "Outdated access point selection for mobile edge computing with cochannel interference," to appear," *IEEE Transaction Vehicular Technology*, vol. 99, pp. 1–10, 2022.

[12] Z. Cai, L. Fu, and W. Li, "Research and analysis of music development based on k-means and PCA algorithm," in *Journal of Physics: Conference Series*vol. 2083, no. 3, IOP Publishing, Article ID 032044, 2021.

[13] Y. M. Jung, J. J. Whang, and S. Yun, "Sparse probabilistic K-means," *Applied Mathematics and Computation*, vol. 382, Article ID 125328, 2020.

[14] A. Azcarraga and F. K. Flores, "A study on self-organizing maps and K-means clustering on a music genre dataset," in *Proceedings of the Theory and Practice of Computation: Proceedings of Workshop on Computation: Theory and Practice WCTP2016*, pp. 219–234, Quezon City, Philippines, September 2018.

[15] J. H. Foleis and T. F. Tavares, "Texture selection for automatic music genre classification," *Applied Soft Computing*, vol. 89, Article ID 106127, 2020.

[16] B. J. D. Sitompul, O. S. Sitompul, and P. Sihombing, "June). Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm," *Journal of Physics: conference Series*, vol. 1235, no. 1, Article ID 012015, 2019.

[17] J. Lu and M. Tang, "Analytical offloading design for mobile edge computing based smart internet of vehicle," *To Appear in EURASIP Journal on Advances in Signal Processing*, vol. 99, pp. 1–10, 2022.

[18] J. Zhao, Y. Fang, Q. Hong, Z. Pan, L. Huang, and D. Zhang, "UAV-based identification of achnatherum splendens community combining K-means and artificial fish swarm algorithm," in *Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2439–2442, IEEE, Yokohama, Japan, 2019 July.

[19] L. Zhang, X. Zhou, and T. Fei, "Research of improved artificial fish swarm portfolio optimization algorithm based on adaptive levy mutation," *Journal of Internet Technology*, vol. 20, no. 6, pp. 1889–1898, 2019.

[20] J. Peng, Z. Wan, Q. Wei, H. Jiang, and S. Wei, "An improved artificial fish swarm algorithm," *International Journal for Engineering Modelling*, vol. 31, no. 4, pp. 57–68, 2018.

[21] A. E. Patil, S. Patil, K. Singh, P. Saraiya, and A. Sheregar, "Online book recommendation system using association rule mining and collaborative filtering," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 11, pp. 83–87, 2019.

[22] L. Zhang and F. Zhu, "DQN based mobile edge computing for smart internet of vehicle," *To Appear in EURASIP Journal on Advances in Signal Processing*, vol. 99, pp. 1–10, 2022.