A music recommendation algorithm based on clustering and latent factor model

Yingjie Jin and *Chunyan* Han*

Software College, Northeastern University, Shenyang, Liaoning Province, China

Keywords: Collaborative filtering, Implicit feedback, Latent Factor Model, K-means.

Abstract. The collaborative filtering recommendation algorithm is a technique for predicting items that a user may be interested in based on user history preferences. In the recommendation process of music data, it is often difficult to score music and the display score data for music is less, resulting in data sparseness. Meanwhile, implicit feedback data is more widely distributed than display score data, and relatively easy to collect, but implicit feedback data training efficiency is relatively low, usually lacking negative feedback. In order to effectively solve the above problems, we propose a music recommendation algorithm combining clustering and latent factor models. First, the user-music play record data is processed to generate a user-music matrix. The data is then analyzed using a latent factor probability model on the resulting matrix to obtain a user preference matrix U and a musical feature matrix V. On this basis, we use two K-means algorithms to perform user clustering and music clustering on two matrices. Finally, for the user preference matrix and the commodity feature matrix that complete the clustering, a user-based collaborative filtering algorithm is used for prediction. The experimental results show that the algorithm can reduce the running cost of large-scale data and improve the recommendation effect.

1 Introduction

In today's information big data era, users' demand for personalized music is constantly increasing, which poses a huge challenge to the intelligent recommendation of music platforms. Currently, frequently used recommendation algorithms include content-based recommendations, collaborative filtering recommendations, and hybrid recommendations. The collaborative filtering recommendation algorithm is one of the most widely used and successful recommendation techniques in the recommendation system. It has the advantage of not relying on the feature information of the project and is not limited by the content analysis technology. This model is currently the most common music recommendation model in implicit feedback scenarios. However, if the number of users is large, this model will produce high dimensional vector calculations in the calculation of large-scale implicit feedback data, which means high computational overhead. Making recommendations

* Corresponding author: hancy0223@126.com

becomes very difficult in this scenario. The model proposed in this paper will use the collaborative filtering algorithm to recommend after user and music clustering to solve the problem of excessive computational overhead.

In recent years, matrix-based decomposition models have gradually replaced traditional relying on neighborhood-based models and become research hotspots and mainstream models. Matrix-based decomposition models have higher prediction accuracy and better scalability. Using matrix decomposition reduces dimension; processing sparse data improves accuracy; parameters can be adjusted for the vector dimension of decomposition, and the complexity of the model does not increase linearly with the increase of the number of users or commodities. However, the traditional matrix decomposition model cannot be recommended in the implicit feedback scenario. The model proposed in this paper is inspired by the matrix decomposition and dimension reduction technique, and uses the latent factor model to represent users and music as low-dimensional vectors. It solves the problem that the training time is too long due to excessive data volume, making it possible to recommend big data sets.

2 Related work

For implicit feedback recommendations based on collaborative filtering, on the one hand, the implicit feedback itself is sparse; on the other hand, due to the lack of negative samples, the model-based method can not learn the information about the negative samples, which will bias the model. In view of the lack of negative feedback and less information in implicit feedback, many scholars have proposed some improved recommendation methods based on implicit feedback. Ruslan et al. [1] proposed a complete Bayesian method for Probabilistic Matrix Factorization (PMF) models, in which model capacity is automatically controlled by integrating all model parameters and hyper parameters. Experiments show that by applying it to the Netflix dataset, the Bayesian PMF model can Be effective trained using the Markov chain Monte Carlo method. Yin et al. [2] proposed the latent factor model IFRM. This model overcomes the difficulty caused by only positive feedback and lack of negative feedback in the implicit feedback recommendation scenario by transforming the recommendation task into the optimization problem of the probability of selection behavior. Yu et al. [3] studied how to use implicit feedback data for personalized recommendation, and proposed an implicit feedback recommendation model that combines context information and user social information. The model proposed by Yu et al. [4] utilized Word2Vec technology in the field of natural language processing. By learning the user's music collection and playing the recorded song co-occurrence information, the user and music can be obtained in a low-dimensional and compact distributed space vector representation. A similarity between the user and the music is established. The online learning algorithm proposed by Wang et al. [5] weakens the habitual behavior and noise of learning users while strengthening the new tendency of learning users. The learning step size is dynamically adjusted for each feedback by comparing the probability of feedback occurrence with the confidence of the user. He et al. [6] proposed a collaborative filtering recommendation algorithm that combines clustering and user interest preferences. According to the user scoring matrix and project type information, the user interest preference matrix is constructed. Then, the K-means algorithm is used to cluster the project set, and then the user is clustered based on the user interest preference matrix. Finally, it is used in each user class cluster. The project score is predicted based on the user's collaborative filtering algorithm.

According to the above research, the latent factor model combined with the matrix decomposition feature and the collaborative filtering recommendation algorithm after clustering can improve the recommendation quality. Based on the above reasons, this paper focuses on the classic problems in the collaborative filtering recommendation algorithm under implicit feedback data environment: excessive data volume and lack of negative feedback. Based on the existing research, this paper has studied the above issues, and the main contributions include:

A music recommendation model combining latent factor model and clustering is proposed to find neighbor users with similar interest preferences to current users, and to cluster users with similar project type interest preferences.

Based on the previous research, the collaborative filtering recommendation algorithm is designed and the parameters are adjusted. Collaborative filtering is performed on user clusters and music clusters, and user-based collaborative filtering recommendation is implemented based on clustering information.

Perform an algorithm comparison experiment on the Million Song dataset.

3 Music recommendation model

We first formally define the recommendation problem based on implicit feedback. The information that the recommendation model can use is the user-music history play record set. When any user i and the list to be recommended L (composed of candidate products to be recommended) are given, the recommendation model can generate the rank L_i of L, which should place the products that the user is more likely to select as possible in front of other products. The problem to be solved is how to construct such a recommendation model based on the historical play record set.

3.1 Model overview

In order to effectively alleviate the negative feedback problem, this paper introduces a large-scale implicit latent factor model. The recorded data of the user listening to music is decomposed into a user preference matrix and a music feature matrix. By fitting the user's choices rather than scoring behaviors, the probability of observable user behavior can be maximized rather than being tailored to a particular evaluation value. In order to effectively alleviate the problem of excessive data volume, this paper proposes a clustering algorithm. The user and music dimensions are clustered separately, and the user interest preference matrix in each project cluster is used to find the neighbor user of the user corresponding to the item to be evaluated. Finally, a user-based collaborative filtering recommendation algorithm is applied to each user class cluster for recommendation.

The specific flow of the collaborative filtering recommendation algorithm for fusion clustering and implicit feedback data proposed in this paper is shown in Fig. 1.

Step1: Firstly through the user-project (User-item) operation record, perform latent factor model operation, obtain user preference matrix U and music feature matrix V;

Step2: Enter the user preference matrix U, and cluster the project dataset by K-means clustering algorithm to obtain k1 user clusters I = {I1, I2, ⋯, Ik1};

Step3: Enter the user preference matrix V, and cluster the project data set by K-means clustering algorithm to obtain k2 user clusters I = {I1, I2, ⋯, Ik2};

Step4: Combines the divided User-item matrix R' with the user cluster to divide the Ik1 music cluster into Ik2, applies the similarity calculation formula in each cluster to find several nearest neighbors of the current user, and then performs score prediction, and Implements Top-N recommendations.

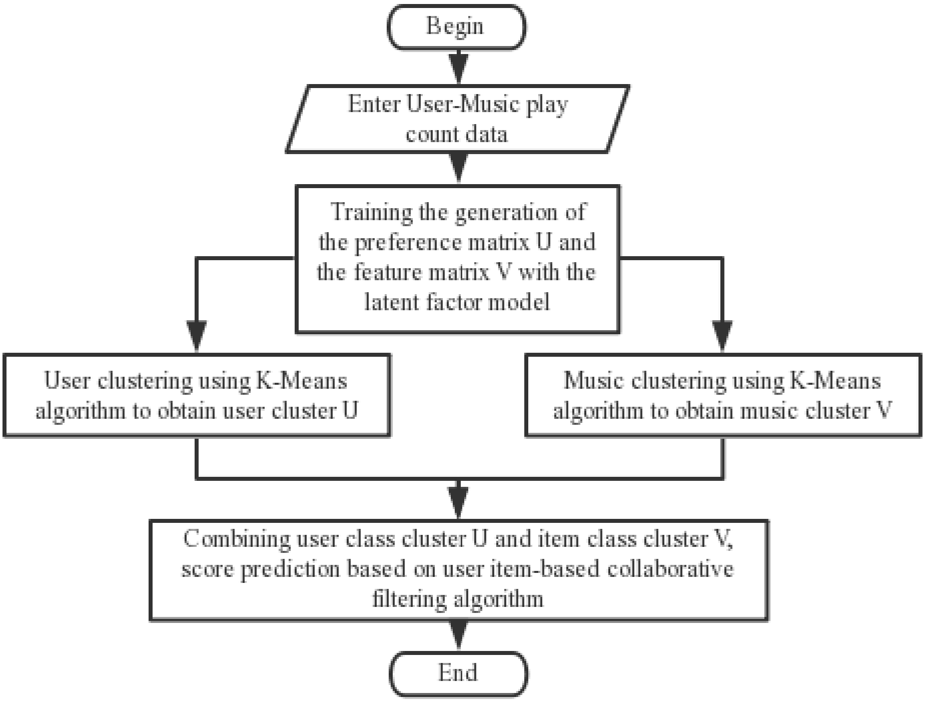
Begin
Enter User-Music play count data
Training the generation of the preference matrix U and the feature matrix V with the latent factor model
User clustering using K-Means algorithm to obtain user cluster U
Music clustering using K-Means algorithm to obtain music cluster V
Combining user class cluster U and item class cluster V, score prediction based on user item-based collaborative filtering algorithm
End

Fig. 1. Algorithm flow.

3.2 Latent factor probability model

Implicit feedback data is characterized only by positive samples, but not negative samples. In order to avoid introducing negative samples during the training process, we use the probability generation model to directly model the user selection behavior. The core is to make reasonable assumptions about the observed reasons for user choice behavior. We assume that the user's choice behavior is determined by the user's "selection tendency" of the product, and the degree of this tendency is relative. In this paper, the reason why the user chooses to listen to music is that the tendency of the user to select the music that has been listened to is higher than that of ordinary music. Based on this, the following formal definitions are first given.

Definition 1. The probability that user i chooses music j is determined by Δ_{ij}. Δ_{ij} describes the relative preference of user i for music j. Δ_{ij} is related to the selection propensity A_{ij} and the average selection propensity $\overline{A}$:

$$\Pr_{ij} = \varphi(\Delta_{ij}) = \frac{\Delta_{ij}}{1+\Delta_{ij}} \tag{1}$$

$$\Delta_{ij} = \frac{A_{ij}}{\overline{A_i}} = \frac{A_{ij}}{\frac{1}{M}\sum_{h=1}^{M} A_{ih}} \tag{2}$$

where M is the total number of products, and $\varphi(x)$ is the Sigmiod function, which is used to normalize Δ_{ij} to the (0,1) interval. The standard S-type function $\varphi(x) = e^x/1 + e^x$ can also be used here. A_{ij} can be considered as a function related to user i and music j, and can be flexibly designed according to information available in a specific application scenario.

Assuming that the observed set of user listening behavior is $\mathrm{O} = \{\langle \mathrm{i}, \mathrm{j}\rangle \mid \text{User i listens to music j}\}$, assuming that the listening behavior is independent of each other, the likelihood probability is:

$$\mathrm{P}(\mathrm{O} \mid \Theta) = \prod_{(\mathrm{i,j})\in \mathrm{O}} \mathrm{Pr}_{\mathrm{ij}} = \prod_{(\mathrm{i,j})\in \mathrm{O}} \frac{1}{1+\Delta_{\mathrm{ij}}{}^{-1}} \tag{3}$$

Among them, generally refers to the model parameters related to the specific design of A_{ij}. Applying the Bayesian formula and a Gaussian prior distribution of the mean 0 and the variance σ^2, the posterior probability can be derived:

$$\mathrm{P}(\Theta|\mathrm{O}) \propto \mathrm{P}(\mathrm{O}|\Theta)\,\mathrm{P}(\Theta) = \prod_{(\mathrm{i,j})\in \mathrm{O}} \frac{1}{1+\Delta_{\mathrm{ij}}{}^{-1}} \mathrm{N}(0, \textstyle\sum_{\Theta}) \tag{4}$$

The purpose of training is to maximize the posterior probability. After taking the logarithm of the equation and negating it, the following optimization goals are obtained:

$$\mathrm{argmin}_{\Theta}\mathrm{L} := \textstyle\sum_{(\mathrm{i,j})\in \mathrm{O}} \ln\left(1 + \Delta_{\mathrm{ij}}{}^{-1}\right) + \lambda_{\Theta} \parallel \Theta \parallel_{\mathrm{F}}^{2} \tag{5}$$

where λΘ is the coefficient of the regular term used to control the complexity of the parameter. Observing the optimization goal, we can see that the latent factor model has the following two advantages: First, it only depends on the user's listening record of music, and does not need negative samples during the training process, so it is naturally applicable to the implicit feedback recommendation scheme. Second, it provides a probability framework, but there is no specific definition. The model parameters are Θ, so they can be generalized. After conversion to an optimization problem, the model parameters can be obtained using the stochastic gradient descent method. Specifically, the user preference matrix U and the music feature matrix V are first initialized. Then, the data in the listening record R is read one by one, while the elements of the corresponding row of the preference matrix are updated in the negative direction of the derivative. Recalculate the selection probability after completing the adjustment. After the selection probability is gradually converged or all the records are trained, the adjusted user preference matrix U and the music feature matrix V are output.

3.3 User and music clustering

The core idea of project clustering is to divide all projects into several clusters according to similarity, and the projects in the same cluster have higher similarity. The purpose of clustering the project set in this paper is to apply the improved collaborative filtering algorithm to the user music matrix in the subsequent processing, and compress the whole project space into several clusters to reduce the impact of operating cost storage noise, and improve the effectiveness of the recommendation results.

K-means and K-medoids are two widely used clustering algorithms. Compared to K-means, K-medoids are computationally more complex and computationally intensive. Therefore, this paper uses the K-means algorithm to cluster projects.

Taking user clustering as an example, the specific steps of the K-means algorithm for clustering the user preference matrix U in the previous article are as follows:

Input: user preference matrix U, number of cluster centers k

Output: K clusters, each with several users, stored as a two-dimensional list

Step 1: Process the user preference matrix U and convert it into a two-dimensional matrix form $\mathrm{U}_{\mathrm{list}}$ that can be directly clustered.

Step 2: Initialize k empty clusters, denoted as $\mathrm{A}\{\mathrm{a}_1, \mathrm{a}_2 \cdots \mathrm{a}_\mathrm{k}\}$

Step 3: Enter the two-dimensional matrix $\mathrm{U_{list}}$ and the number of cluster centers k, and then run the K-means algorithm to get a one-dimensional list L. Each element $\mathrm{L_i}$ in the list represents a cluster assigned to the user $\mathrm{U_i}$ by the clustering algorithm.

Step 4: According to the user allocation list L, each empty cluster $\mathrm{a_i}$ is assigned a corresponding user element $\mathrm{U_j}$. Finally, the cluster output is used for the next step.

The value of the number K of project clusters is critical. Because when the K value is too small, it is usually impossible to achieve efficient division of the project set, and there are fewer similar projects and noisy project elements in the cluster, which brings errors to the algorithm prediction. When the K value is too large, the number of elements in the cluster is too small, and the item data of the scoring reference is insufficient, which also affects the accuracy of the algorithm.

3.4 Collaborative filtering recommendation

There are three similarity calculation methods commonly used in collaborative filtering: cosine similarity, modified cosine similarity, and Pearson correlation similarity.

In this paper, cosine similarity is used for calculation convenience. The similarity between two users can be regarded as the cosine of the angle between the two user score vectors. The larger the cosine value, the higher the similarity between users. Assuming that the scoring vectors of two users x and y are u and v, respectively, the cosine similarity $\mathrm{sim(u,v)}$ between x and y is:

$$\mathrm{sim(u,v)} = \mathrm{cos(u,v)} = \frac{\mathrm{u \cdot v}}{\|\mathrm{u}\| \cdot \|\mathrm{v}\|} \tag{6}$$

The K-nearest neighbor calculation based scoring prediction and recommendation usually adopts the K-nearest neighbor method for scoring prediction, that is, K users who are most similar to the target user are selected as the nearest neighbor set for calculation. Assuming that the set U represents the nearest neighbor set of the target user u, the predicted score value of the user u for the item i is:

$$\mathrm{p(u,i)} = \overline{\mathrm{R_u}} + \frac{\sum_{\mathrm{u_k \in U}} \mathrm{sim(u,u_k)} \times (\mathrm{R_{u_k,i}} - \overline{\mathrm{R_{u,k}}})}{\sum_{\mathrm{u_k \in U}} \mathrm{sim(u,u_k)}} \tag{7}$$

After the score prediction is completed, the top N items that have the highest predicted score and are not in the target user's evaluated item set are taken as the Top-N recommendation set, and the recommendation is implemented.

In this paper, the establishment of the data matrix is performed according to the clusters obtained in the foregoing. Firstly, a two-dimensional matrix is established for the user preference matrix and the music feature matrix based on the cluster classification. Each row of the matrix represents a cluster, and the potential feature elements of each cluster are the average of all users or music features in the cluster, and two cluster numbers feature dimension matrices are obtained. Then, the two matrices are matrix multiplied to obtain the user cluster music class cluster scoring matrix. Thus the original data is divided into several blocks. Then, using the obtained scoring matrix, the matrix element values are adjusted and recommended using a user-based collaborative filtering algorithm. Finally, the recommendation is implemented in the recommended block.

4 Experiments and results

This section describes the test environment and the experimental content. The experimental environment includes the environment and dataset in which the code runs and the evaluation metrics. Relevant experimental content: Determination of relevant parameters in the algorithm; comparison with the comparison algorithm to see if there is any significant improvement.

4.1 Experimental environment and dataset

The experimental environment of this article is a PC, the operating system is Windows 7, and all the programs in the experiment are implemented in Python. This article selects the Million Songs Dataset as the experimental test dataset.

The Million Songs Dataset was created with funding from the National Science Foundation (IIS) project IIS-0713334. The original data was provided by Echo Nest and is part of the NSF-sponsored GOALI collaboration. The core of the dataset is The Echo Feature analysis and metadata for a million songs provided by Nest.

The Echo Nest Taste Profile Subset data set was selected in the experiment, which included 1,019,318 users' 48,373,586 listening records for 384,546 unique MSD songs. In the experiment, the data set is randomly divided into two parts: the training set and the test set, in which the training set accounts for 80% of the entire data set and the test set accounts for 20%.

4.2 Experimental results and analysis

The prediction accuracy measures the approximation between the predicted score and the true score. In order to examine the gap between the predicted value and the true value, this paper uses RMSE to measure the accuracy of the collaborative filtering algorithm. The smaller the RMSE value, the higher the accuracy of the algorithm.

In order to verify the effectiveness of the proposed algorithm, there are three stages in the experiment.

Experiment 1: Effect of Cluster Number K on RMSE

This paper needs to cluster the project dataset to obtain several project clusters, so that the projects in the same cluster have higher similarity. In this experiment, the range of K is 20 to 60, and the interval between each time is 5, and the influence of the change of K on the recommendation accuracy is observed in turn. Finally, the optimal K value is selected, and the experimental results are shown in Fig. 2. It can be seen that when K is 50, the RMSE of the recommended algorithm is the smallest. Therefore, in this paper, when clustering the project, the clustering cluster number K is 50.

Experiment 2: Effect of User Feature Dimension D on RMSE in Latent Factor Model

In this experiment, the feature dimension is selected from 30 to 70, and the interval is 10. The influence of the change of the feature dimension D on the accuracy of the algorithm recommendation is observed in turn, and the value of the feature dimension D when the best recommendation effect is selected. The experimental results are shown in Fig. 3. The RMSE of the change of the feature dimension D is experimentally known. When the number of neighbors taken is around 30, the RMSE of the prediction score of this algorithm reaches the lowest, that is, the recommended effect of the algorithm is the best, so the number of neighbors is D=30.

Experiment 3: Comparison of Recommended Accuracy with Other Algorithms

In order to evaluate the recommended accuracy of the algorithm in this paper, this experiment compares it with three other recommendation algorithms, including the traditional user-based collaborative filtering algorithm (User-base CF), project-based collaborative filtering algorithm (Item-base CF) and the latent factors recommended algorithm proposed in the literature. In this experiment, the number of clusters of project K-30 is selected. For each algorithm, the cosine similarity formula is used to calculate the user similarity, and the influence of different algorithms on the recommendation accuracy is observed in turn. The experimental results are shown in Fig. 4.

5 Conclusion

Aiming at how to improve the accuracy and efficiency of personalized recommendation methods in the era of big data, this paper studies the efficient recommendation algorithm design based on large-scale implicit user feedback data and the clustering collaborative filtering algorithm. A novel hybrid recommendation algorithm c-IFRM combining clustering and latent factor model is proposed. The clustering algorithm is introduced into the latent factor model, and the cluster classification information is used to further improve the recommendation result for the target users. Experiments show that the c-IFRM model is used to provide users with personalized recommendations, which are more widely applicable and have higher recommendation accuracy. However, the c-IFRM model causes the algorithm to gradually diverge at a later stage without converge by using a large amount of user music to play data. This problem can be solved by sampling an appropriate amount of data from the original data to ensure a certain degree of convergence, or by changing the learning rate to correct the convergence process of the algorithm. Our next research work will focus on the combination of latent factor models and other machine learning algorithms. See if you can find other combinations that are more suitable for implicit feedback data environments.

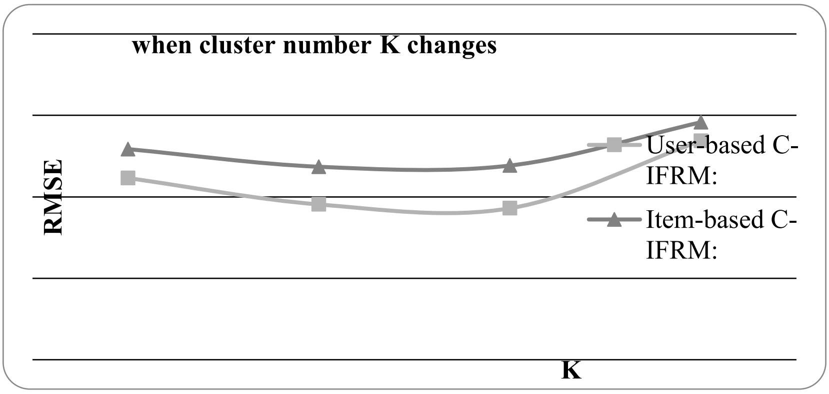

Fig. 2. RMSE when cluster number K changes.

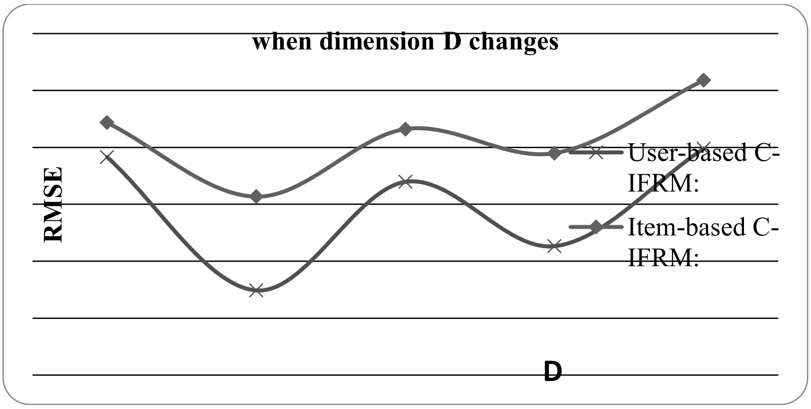

Fig. 3. RMSE when dimension D changes.

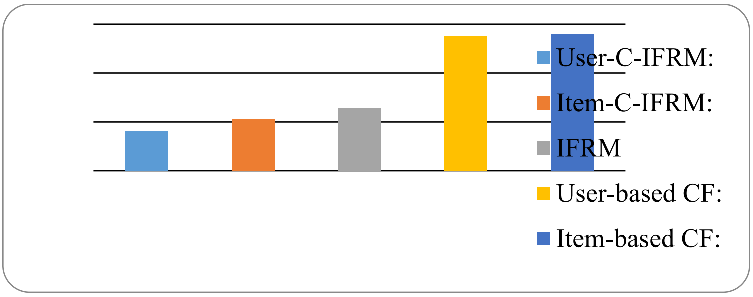

Fig. 4. Comparative Experiment.

References

1. Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using markov chain monte carlo[C]// International Conference on Machine Learning. ACM, 2008.
2. Yin J, Wang ZS, Li Q, Su WJ. Personalized recommendation based on large-scale implicit feedback. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):1953−1966 (in Chinese). http:// Www.jos.org.cn/1000-9825/4648.htm
3. Yu Chunhua, Liu Xuejun, Li Bin, et al. Context-aware recommendation of social information fusion in implicit feedback scenarios[J]. Computer Science, 2016, 43(6): 248-253.
4. Yu Shuai, Lin Xuanxiong, Qiu Yuanyuan. A Word Vector Music Recommendation Model for Large-Scale Implicit Feedback. Computer Systems Applications, 2017, 26(11): 28–35. http://www.csa.org.cn/1003-3254 /6049.html
5. Wang Zhisheng, Li Qi, Wang Jing, et al. Real-time personalized recommendation based on implicit user feedback data stream [J]. Chinese Journal of Computers, 2016(1): 52-64.
6. He Ming, Sun Wang, Xiao Run, et al. A Collaborative Filtering Recommendation Algorithm Based on Fusion Clustering and User Interest Preference [J]. Computer Science, 2017(S2): 401-406.
7. Li Tao, Fu Ding. Automated Implicit Grading Music Dual Recommendation System Based on Collaborative Filtering Algorithm [J]. Journal of Computer Measurement and Control, 2018, 26(11): 177-181.