Hindawi
Complexity
Volume 2021, Article ID 6696986, 11 pages
https://doi.org/10.1155/2021/6696986

Research Article

Automatic Recommendation Algorithm for Video Background Music Based on Deep Learning

Hong Kai

Department of P. E. and Art Education, Zhejiang Yuexiu University, Zhejiang, Shaoxing 312000, China

Correspondence should be addressed to Hong Kai; 20142039@zyufl.edu.cn

Received 31 December 2020; Revised 15 January 2021; Accepted 23 January 2021; Published 3 February 2021

Academic Editor: Wei Wang

As one of the traditional entertainment items, video background music has gradually changed from traditional consumption to network consumption, which naturally also has the problem of information overload. From the perspective of model design and auxiliary information, this paper proposes a tightly coupled fusion model based on deep learning and collaborative filtering to alleviate the problem of poor prediction accuracy due to sparse matrix in the scoring prediction problem. In the use of auxiliary information, this paper uses crawler technology to obtain auxiliary information on the user side and the video background music side and compensates for the model's sensitivity to the sparsity of the score matrix from a data perspective. In terms of model design, this paper conducts auxiliary information mining based on the diversity and structural differences of auxiliary information, uses an improved stack autoencoder to learn user's interests, and uses convolutional neural networks to mine hidden features of video background music. Based on the idea of probabilistic matrix decomposition, the tightly coupled fusion of multiple deep learning models and collaborative filtering is realized. By comprehensively considering user's interest and video background music characteristics, the collaborative filtering process is supervised, and the optimized prediction result is finally obtained. The performance test and function test of the system were carried out, respectively, to verify the effectiveness of the hybrid recommendation algorithm and the effect of the system for recommendation. Through experimental analysis, it is proved that the algorithm designed in this paper can improve the recommendation quality and achieve the expected goal.

1. Introduction

As people's ability to collect video background music and store video background music becomes stronger and stronger, users' preferences for video background music gradually have their own style, which provides great difficulty for the preference recommendation function of music listening software [1, 2]. Although some manual and simple mining methods have been used to improve the accuracy of the system's recommended songs for users, the analysis and processing capabilities of the structure of the video background music itself have not been correspondingly improved. Regarding the connotation of the video background music itself, perhaps only a video background musician or a professional music critic can truly understand the meaning of the video background music and can distinguish the type of the video background music when listening to the video background music [3–5]. However, for ordinary users, these important pieces of information can only roughly hear the mood and themes to be expressed, most of which are passive perceptions. Only artists can understand the isomorphic correspondence between the sound waves of the video background music and human emotions. This makes humans have limitations in understanding video background music, and there are individual differences. With the development of computer technology, its application has spread to all fields of economic and social life. Therefore, discovering the isomorphic relationship between video background music and human emotions through computers has become a means to solve such problems [6, 7].

For the recommended video background music system, different video background music has different styles, divided into many video background music genres and video background musicians, and different video background

music genres of the same video background music are understood by video background music. Among the many video background music listeners, each user has his own unique hobby [8]. This hobby is often reflected in the process of users listening to the video background music. Therefore, it is necessary to use the video background music recommendation system service to recommend their favorite video background music [9]. The attribute information of the video background music refers to the singer, author, style, album, format, and song time of the video background music. Using this information, the video background music can be clustered, the similarity of the video background music can be calculated, and then a recommendation list can be generated. The essence is to turn the video background music into a vector, and each dimension is composed of its attributes and its proportions [10]. Various similarity measurement methods are used to calculate the similarity, and finally the results are recommended to users. However, this recommendation method has certain disadvantages, because the attribute information of the video background music does not fully reflect the video background music itself [11]. For example, the video background music in the same album may also have different styles, so the effect of this recommendation method is not very good. The idea of hybrid recommendation takes into account information other than the video background music itself. For example, external information such as the listening environment can be used, because when the user is listening to the video background music, the mood is related, and environmental factors may affect the mood of the user at that time [12]. After collecting this external information, we cluster the video background music to obtain different types of video background music (happy, sad, etc.) and then recommend them to users with different needs [13]. However, this method is a relatively blunt recommendation method, which is not suitable for personalized recommendation, and the introduction of external information will increase the calculation workload and will also affect the playback effect of the video background music, resulting in poor quality of the recommendation results [14]. Related scholars published a paper on a deep neural network coded by a restricted Boltzmann machine [15]. The paper proposed the use of restricted Boltzmann machines to pretrain hierarchically and then use real data sets for parameter adjustment. This paper brings Boltzmann machines and neural networks back to the sight of academia. Related scholars have proposed a personalized search ranking algorithm based on local knowledge bases [16]. The degree of preference extracts typical documents from categories, and typical documents from different categories constitute the user's personalized local document library. The final search results are sorted according to the degree of similarity with the local document library. Related scholars have proposed a recommendation system based on Deep Belief Network (DBN) [17]. The friendliness of deep learning to big data, strong representation ability, and excellent antinoise ability all make deep learning more attractive in the recommendation field. But deep learning is not a panacea. First of all, deep learning-based model training costs are extremely high, require a lot of data, and consume a lot of energy. Secondly, the abstract representational features extracted by the machine are not easy for people to understand. For resources with a small amount of data and easy feature extraction, traditional recommendation methods balance cost and output [18]. Based on the traditional model, researchers have proposed a new method that comprehensively considers the total number of tag citations and the total number of tags in the configuration file [19]. It is mainly used to calculate the weight of tags in the user profile and resource configuration file, and the calculation method matches the query and resource profile and user profile [20, 21]. Early research on collaborative filtering methods in the field of video background music is based on explicit feedback based on the user's rating of songs or artists [22]. However, since the most common feedback collection method currently is to collect user listening records, the feedback form used in the field of video background music recommendation has changed from explicit to implicit. The use of implicit feedback has obvious disadvantages; that is, users no longer express their preferences for songs explicitly, but infer their preferences from the records of their listening to songs [23, 24].

This paper introduces the proposed tightly coupled deep learning fusion model in modules. It introduces the principle and process of mining hidden features on the user side based on the extended stacked autoencoder and then introduces the mining of hidden features on the video background music side in detail, from the representation of text data to the use of deep learning models. Specifically, the main contributions of this paper can be summarized as follows:

First, in terms of model design, this paper uses convolutional neural networks to mine the hidden features of video background music and improves this model by introducing an attention mechanism. The local key points of the text can better locate the local keywords and improve the accuracy of the hidden feature mining on the video background music side. The two content mining-based deep learning models are tightly coupled through the probabilistic matrix decomposition model, and the feasibility and operating mechanism of the model are explained theoretically through formula derivation.

Second, according to the characteristics of the digital video background music service platform, this article uses social tags to describe the extra information of the video background music, designs a method that can infer the context and recommends, and implements a system on this basis. The system can provide services similar to the internet radio stations and complete the evaluation of the effectiveness of the recommendation algorithm through interaction with users.

Third, this paper verifies the effectiveness of the deep learning algorithm and the recommendation effect of the system based on this algorithm and shows from the

results that the recommendation algorithm designed by this paper can improve the recommendation quality.

The rest of this article is organized as follows. Section 2 discusses the related theories of automatic recommendation of video background music. Section 3 constructs a recommendation algorithm based on the deep learning fusion model. In Section 4, experimental testing and result evaluation were carried out. Section 5 summarizes the full text.

2. Video Background Music Automatic Recommendation Related Theories

2.1. Video Background Music Recommendation System. In this information overloaded Internet era, how to make users make full use of information on the internet has always been a hot research direction and research pain point. In this environment, the cost of internet information producers who want to make their information attract people's attention in a large number of information streams is getting higher and higher; it is also difficult for consumers to find what they really need in the vast ocean. Mature business solutions are mainly divided into search engines and recommendation systems. The recommendation system is based on the user's historical behavior to dig out potential points of interest. Although search engines and recommendation systems have different ways of conveying information, they also have similarities in their differences. The common purpose of both is to help users obtain more valuable information, and the difference lies in whether users actively obtain or passively recommend. The overall architecture of the video background music recommendation system is shown in Figure 1.

The recommendation system is a typical data-driven product. In order to realize an effective recommendation service provider, it is necessary to collect different information of users or video background music to characterize it. There are many types of data sources used in the recommendation system, which can be roughly divided into two categories: one is content-related data describing users and video background music, and the other is data generated by interaction between users and video background music. According to the different use of data sources, the current general recommendation system algorithm classification method is generated. If the recommendation system uses content-related data of users or video background music, the recommendation system is considered to be a content-based recommendation system.

2.2. Collaborative Filtering Algorithm. The recommendation model based on matrix decomposition assumes whether the user likes a piece of video background music. The reason behind it is controlled by a series of potential factors with different weights. The corresponding background music of each video also has a different set of potential factors. If the potential factor of the user and the potential factor of the video background music are innerly produced, a quantitative value of the user's preference for the video background music can be obtained. For each user, it need to calculate the quantitative value of the background music in all the videos that have not been rated and then sort them from largest to smallest, and then the user's prediction recommendation list can be obtained to realize the recommendation function. The entire calculation process is performed offline, so the online prediction recommendation process becomes very fast, and the energy efficiency of the model-based collaborative filtering algorithm is usually better than that of the neighbor-based collaborative filtering algorithm.

The latent factor model decomposes the rating matrix R into a user hidden feature matrix U and a video background music hidden feature matrix V. Each row vector in the U matrix is the hidden feature vector of user i, and u_{ik} is the weight of the user on the k-th latent factor. Each row vector in the V matrix is the hidden feature vector of the video background music j, and the same v_{jk} is the weight of the video background music on the k-th latent factor. The inner product of the two vectors represents the "preference" of user i on the video background music j, and the public expression is as follows:

$$r_{ui} = \sum_{k=0}^{K-1} u_{ik} \times v_{jk}. \tag{1}$$

Among them, K represents the number of latent factors, that is, the dimensions of the u_i vector and vj. r_{ui} is the value predicted by the recommendation system. In order to obtain two characteristic matrices, LFM defines the following loss function to calculate U and V matrices:

$$L(U, V) = \lambda \cdot \left[|U|^2 + |V|^2\right] + \prod_{i=0}^{M-1} \prod_{j=0}^{N-1} I_{ij} \cdot \left(U_i^T V_j - R_{ij}\right)^2. \tag{2}$$

In the latter half of the loss function, a regularization term is added to prevent the model from overfitting. λ is the parameter of the regularization term, and I_{ij} is the indicator function.

At the same time, the difference in the rating interval of users is taken into consideration. For example, some users are accustomed to giving high scores regardless of whether the background music of the video is good or bad, while some users are very extreme, and the background music of the video they like is highly rated. Therefore, when predicting the score, the preference information of the user and the video background music is introduced.

$$r_{ui} = u_i^T v_j + b_i + b_j + u. \tag{3}$$

Among them, u is the average value of nonzero elements in the matrix, b_i is the average rating of user i, and b_j is the average rating of video background music j.

The loss function becomes

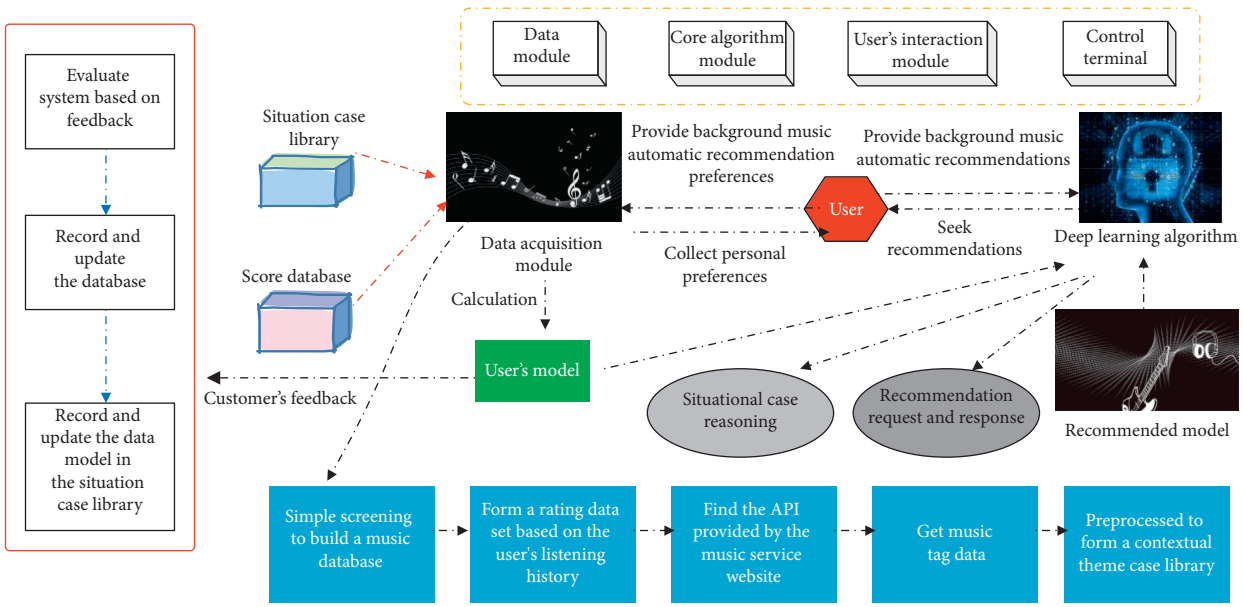

Figure 1: Overall architecture of the video background music recommendation system.

$$L(U, V) = \lambda \cdot \left[\sum_{i=0,j=0}^{M-1,N-1} \left(b_i^2 \cdot b_j^2 \right) + \|U\|^2 + \|V\|^2 \right] + \prod_{i=0}^{M} \prod_{j=0}^{N} I_{ij} \left(b_j + b_i + u + U_i^T V_j - R_{ij} \right)^2. \tag{4}$$

The optimization method of the loss function is usually a gradient descent method. The gradient descent algorithm updates the parameters by obtaining the partial derivatives of all unknown parameters in the loss function. The negative gradient direction of the parameters is the direction in which the loss function drops the fastest. Updates are continuously performed through iterations, and iterating stops until the set number of iterations or the reduction of the loss function is lower than the set threshold. Assuming that the learning rate in gradient descent is θ, the parameter update formula in each round of iteration is

$$\begin{aligned} u_{ik} + \theta \left(e_{ij} v_{jk} - \lambda u_{ik} \right) &\longrightarrow u_{ik}, \\ u_{ik} + \theta \left(e_{ij} v_{jk} - \lambda u_{ik} \right) &\longrightarrow u_{ik}, \\ b_i + \theta \left(e_{ij} - \lambda b_i \right) &\longrightarrow b_i, \\ b_j + \theta \left(e_{ij} - \lambda b_j \right) &\longrightarrow b_j. \end{aligned} \tag{5}$$

After the update, the model outputs the user hidden feature matrix U and the video background music hidden feature matrix V. In the prediction stage, the prediction score can be obtained by the vector inner product of the user and the song in the matrix.

2.3. Audio Content Characteristics. The first step of MFCC calculation is to use a filter bank composed of triangular filters to convert the amplitude spectrum obtained by DFT into a mel scale. Each triangle filter defines the response of a frequency band and normalizes it so that the sum of the weights of each triangle is the same.

A filter bank $F_i(k)$ is composed of M triangular filters of equal height, and the filter of each filter is defined as

$$F_i(k) = \begin{cases} 0, & f_{b_i+1} < k, \\ \dfrac{\left(f_{b_i+1} - k \right)}{\left(f_{b_i+1} - f_{b_i} \right)}, & f_{b_i+1} \geq k \geq f_{b_i}, \\ \dfrac{\left(f_{b_i-1} - k \right)}{\left(f_{b_i-1} - f_{b_i} \right)}, & f_{b_i-1} \leq k \leq f_{b_i}, \\ 0, & k < f_{b_i-1}. \end{cases} \tag{6}$$

Among them, i represents the i-th filter, and f_{bi} is the boundary point of the filter, corresponding to the k-th coefficient of the N-point DFT. The position of the boundary point f_{bi} depends on the sampling frequency Fs and the point N in the DFT.

The number of filters is equal to the number of MFCC coefficients. Then through Discrete Cosine Transform (DCT) to obtain the MFCC coefficients, its physical meaning is the distribution of the energy of the signal spectrum in different frequency intervals. The function of each filter is to obtain the spectral energy of the corresponding frequency range. MFCC coefficient calculation formula is

$$C_j = \sum_{i=0}^{M-1} X_i \cdot \cos \left[\frac{\pi}{M-1} \cdot j \cdot (i - 0.5) \right], \tag{7}$$

where M is the number of filters in the filter bank, j is the number of cepstral coefficients ($j < M$), and X_i is the logarithmic energy output of the i-th filter; the formula is

$$X_i = \log_{10}\left[\sum_{k=0}^{N-1} X(k)|F_i(k)\right], \tag{8}$$

where $X(k)$ is the amplitude spectrum of the Fourier transform. Through these two steps, the audio can be described by a series of cepstrum vectors; each vector is the MFCC coefficient of each frame.

The spectral centroid is a metric used to characterize the frequency spectrum in digital signal processing. It shows where the "centroid" of the amplitude spectrum of the short-time Fourier transform is and measures the average frequency of the amplitude-weighted spectrum. The brightness of the timbre in human perception is related to this feature. Its formula is

$$\text{Centroid}(X) = \frac{\sum_{n=0}^{N-1} n \cdot X(n)}{\sum_{n=0}^{N-1} X(n)}. \tag{9}$$

Among them, $X(n)$ is the amplitude of the Fourier transform at frequency n, and X is a DFT frame. N is the number of frequency points, for example, half of the number of samples in the DFT frame.

The main purpose of the rhythm feature extraction is to extract the regular change features in the time sequence of the audio, such as rhythm, beat, and rhythm structure. Rhythm is the speed of a piece of music, and human intuitive feeling is the speed of the song. In a given prosodic structure, the rhythm is the beat rate in the prosodic structure, and the beat describes the time when an acoustic event occurs. The rhythm structure describes the basic law of the occurrence of video background music events.

The calculation of rhythm features is based on the periodicity of the measured things. In video background music, the measured things refer to the starting point in the audio file and the extracted low-level features (such as energy, spectral features, etc.). Specifically, the starting point is used to estimate the beat position. For example, the starting point in a piano song is the moment when the key is pressed. When the starting point is combined with the spectral characteristics, the loudness at the current point can be estimated.

3. Recommendation Algorithm Based on Deep Learning Fusion Model

3.1. User Interest Mining Based on Extended Stack Autoencoder. Through the mining of user-side content information, the user's hidden feature matrix can be obtained, which can be regarded as an implicit factor matrix that affects the user's preference for video background music. Usually, the user-side content information that can be obtained is mostly structured short-text information, such as user's gender, age, occupation, zip code, etc. This information does not involve word order issues. Based on the previous introduction and investigation and analysis, it is found that the SDAE model is used for structure. When mining user's interests, this paper expands the SDAE model (abbreviated as ASDAE), adds user's historical behavior feedback information when inputting the model, and restricts the reconstruction and training process of the model, so that the model is encoded. The obtained feature representations are more abundant, and the effect of the ASDAE model is found to be improved to a certain extent compared with the SDAE after testing. The basic idea of the model is to learn the hidden features of the input data from the noisy input data and obtain the dimensionality reduction feature that reexpresses the original data by minimizing the reconstruction error. The model includes an encoder and a decoder. Assuming that an SDAE contains L layers, the first $L/2$ layer is the encoder, and the last $L/2$ layer is the decoder. In this paper, the user's interest mining process based on the extended stack autoencoder is shown in Figure 2.

Tonal feature is an important part of video background music, so it is very important to extract the tonal feature content from the song to characterize the video background music content. In video background music, pitch (also known as tonality) is the general term for the main tone and mode category of the key. Tone can be thought of as a series of different musical tones around a tonic. In addition to the tonic, there are two important pitches in the tune: dominant and subordinate. Mode is to organize music tones together according to a certain interval relationship and become an organic system. According to the difference in the arrangement structure of the interval relationship, the modes can be divided into two categories: major and minor. For example, in the tonal feature of a song, the tonic is C and the interval relationship arrangement is a major; then the tonal feature of the song can be described as "C major." Generally speaking, the tonal characteristics of a song determine the most intuitive emotional expression that the song brings to the listener. It is generally believed that major songs can give listeners a broad and bright feeling, while minor songs can give listeners a feeling of lyric and melancholy.

The specific user's interest mining process is as follows:

(1) First, you obtain and encode user-side content information and convert the structured content information on the user-side (such as ID, age, gender, occupation, etc.) into vector form X mainly through one-hot method.

(2) You convert the user's historical behavior information (scoring matrix) into a user feedback table; that is, set the interactive position to 1, and set the noninteractive position to 0, and then obtain the user's feedback set for all video background music.

(3) The greedy method is used to pretrain each layer of the ASDAE model layer by layer to complete the parameter initialization of the ASDAE network. When performing layer-by-layer training, for each layer in the network, an output layer is added first,

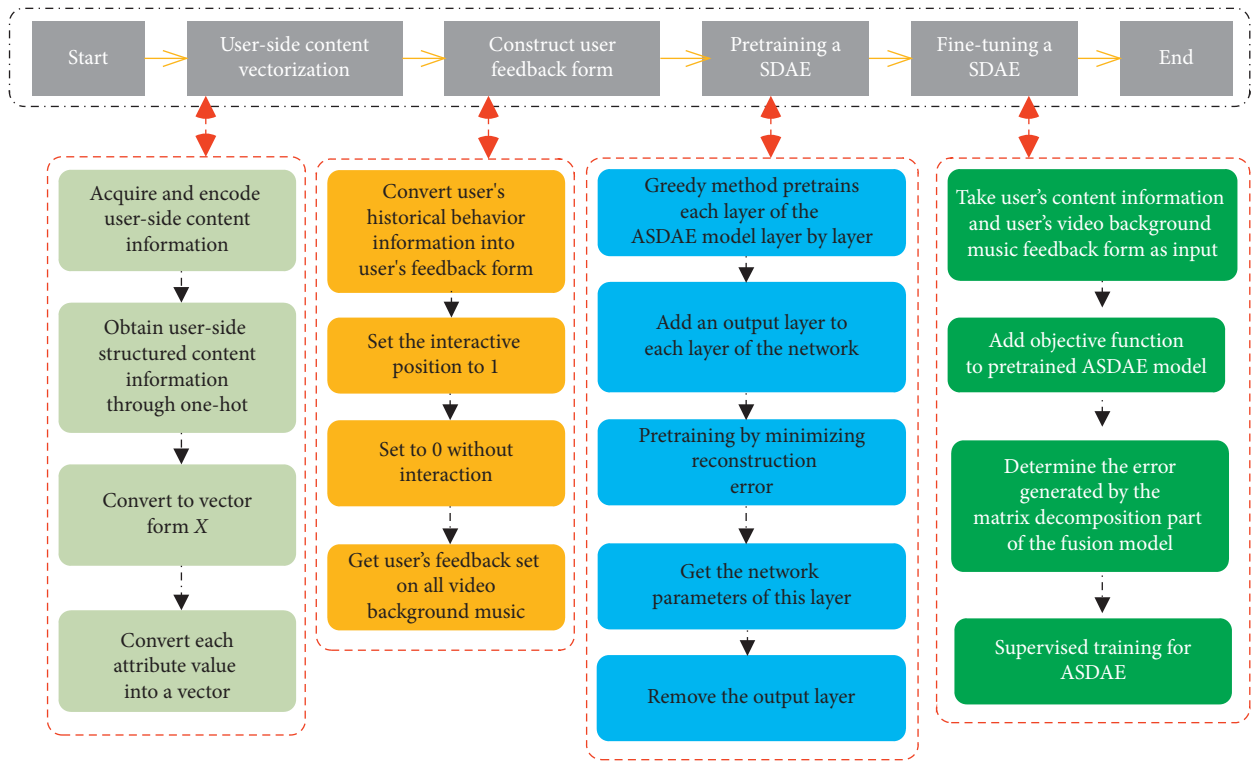

Figure 2: Flowchart of user interest mining based on extended SDAE.

and then pretraining is performed by minimizing the reconstruction error, and finally the network parameters of this layer are obtained, and then the output layer is removed. The function of the coding layer is as follows:

$$f(l) = (g + b_l) \cdot (W_l h_{l-1} + XV)_l. \tag{10}$$

(4) You take the vectorized user content information X and the user video background music feedback table S as input, input them into the model ASDAE, and add an objective function to the pretrained ASDAE model, and supervise and train the ASDAE to fine-tune the network parameters. The objective function here should be determined according to the task of the model, here is the error between the user implicit matrix generated by the matrix decomposition part of the fusion model and the user interest matrix output by the model.

3.2. Video Background Music Feature Mining Based on Extended Convolutional Neural Network. In the long-text information on the video background music side, the comment information not only contains some attributes of the video background music, but also contains the user's emotional tendency, which is rich in information, and relatively speaking, the extracted features may be more valuable. Therefore, when this article introduces the content information on the video background music side, this article finally decided to add comment data. In the text mining problem, there are many attempts to apply the CNN model, which is popular because of its simplicity and good effect. The general idea of using the CNN model is to first vectorize the entire text and then use convolution, pooling, and fully connected layers to gradually extract text features. But for a long text, such as the comment information used in this article, a piece of information may contain the functions, characteristics, and user's attitudes of the video background music. The importance of the text in different areas in the entire text is not equivalent. In fact, the text is full of a lot of words that have nothing to do with the attributes of the background music of the video. The grasp of the key words in the local area is conducive to more accurately obtaining the attributes of the background music of the video. Therefore, this article expands the CNN model and introduces the attention mechanism into it, which is recorded as ACNN. Through the following experimental comparison, it is found that the improved model is helpful to the improvement of the effect, so it also shows the correctness of the introduction of the attention mechanism. The structure of the improved CNN model is shown in Figure 3.

The convolutional layer is used to extract text features. Due to the particularity of the contextual information contained in the text, it is different from computer signal processing in terms of processing. Therefore, after obtaining the text sequence with attention effect, it is necessary to further extract the document features through the convolution structure. The working principle of the convolutional layer is to uniformly transform the input image or text into the form of a numerical matrix and use a fixed-size convolution kernel to roll the numerical matrix from left to right and top to bottom in a certain step. The convolution kernel

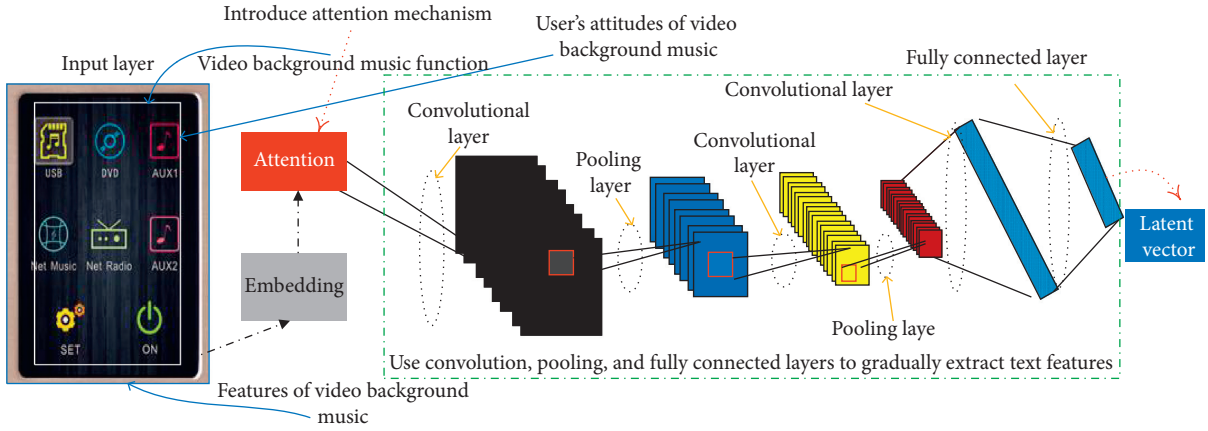

Figure 3: Structure diagram of video background music feature mining based on ACNN.

can be understood as a weight matrix, and the convolution operation is to take the sum of the product of the convolution kernel and the number at the corresponding position in the process of moving as the result.

Let W_c denote the weight matrix of the convolutional layer and b_c denote its bias information, then the text feature C obtained after convolution is shown in the following formula. Among the nonlinear activation functions, there are sigmoid, Re LU, etc. After comparing the effects, this article uses the better Re LU activation function to avoid the problem of gradient disappearance, which is represented by $*$ in the above formula.

$$C(i) = f\left(b_c * W_c + T_i^A\right). \tag{11}$$

The pooling layer further excavates the text features obtained after convolution; while ensuring that the target is not affected by relative position changes, it also reduces the dimensionality of the extracted text features, which can avoid overfitting to a certain extent. The working principle of the pooling layer is to set a fixed-size volume core for the feature map obtained in the upper layer and perform a pooling operation from left to right and top to bottom according to a certain step. There are two ways of pooling operation, taking the maximum or average value in the corresponding range of the convolution kernel without going through the back-propagation process.

The fully connected layer is used to synthesize the previously extracted features and map the synthesized result to a vector space of a specific dimension as the final output of this part of the model. Each node in the fully connected layer will be connected with all the outputs of the previous layer to achieve the purpose of feature mapping. Through full connection, you can finally get text features that meet the requirements of specific dimensions. The feature output obtained after the model passes through the fully connected layer is

$$C_{\text{out}} = g\left(C_{\text{pool}} + b_{\text{FC}} \cdot W_{\text{FC}}\right). \tag{12}$$

3.3. Recommendation Algorithm Based on Deep Learning Fusion Model. According to the principle of PMF, first, we assume that the two implicit characteristic matrices U and V obtained after decomposition all obey Gaussian distribution. Based on the Bayesian formula and the knowledge of the maximum posterior probability, the problem of maximizing the log-posterior probability can be transformed into the problem of minimizing the loss function. During model optimization, the parameters are updated alternately. V is updated by fixing U and W, or U is fixed by fixing V and W. Through the gradient descent method, the update method of U and V can be obtained:

$$\begin{aligned}\left(\lambda I_k - \text{VI}_i V^T\right)^{-1}\left(\text{VR}_i - \lambda h_i\right) &\longrightarrow u_i,\\ \left(\lambda I_k - \text{UI}_j U^T\right)^{-1}\left(\text{UR}_j - \lambda C_{out}\left(W, D_j\right)\right) &\longrightarrow v_j.\end{aligned} \tag{13}$$

Finally, after obtaining the hidden feature matrix U and V, the score prediction is made by the following formula:

$$R_{ij} = u_i \cdot v_j^T. \tag{14}$$

Through the gradient descent method, the update method of V can be obtained as

$$\text{UR}_j \cdot \left(\lambda I_k - \text{UI}_j U^T\right)^{-1} \longrightarrow v_j. \tag{15}$$

Through the gradient descent method, the update method of U can be obtained as

$$\text{VR}_i \cdot \left(\lambda I_k - \text{VI}_i V^T\right)^{-1} \longrightarrow u_i. \tag{16}$$

4. Experimental Test and Result Evaluation

4.1. Performance Test. Since the algorithm designed in this paper involves the selection of similarity measurement methods in the first stage, it is necessary to evaluate the effect of collaborative filtering under different measurement methods on the data set. For the collaborative filtering stage using the deep learning fusion model and the Euclidean metric method, the comparison of the MAE values under different proportions of the training set is shown in Figure 4. It can be seen that the prediction scoring accuracy of the deep learning fusion model is generally higher than that of the Euclidean similarity. In the case of using a complete data set for actual recommendation, it should be more accurate to use a deep learning fusion model to measure the user's predicted preference models.

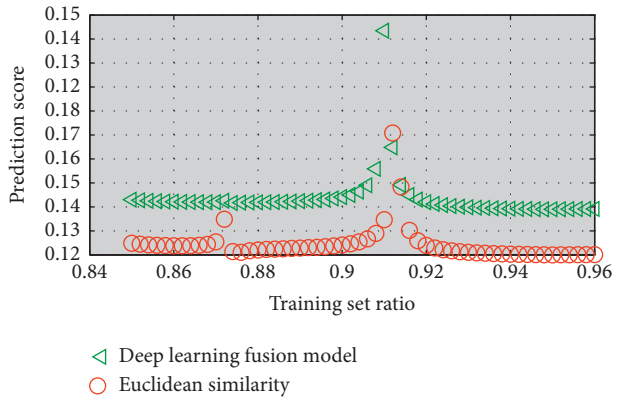

Figure 4: Comparison of prediction score accuracy using Euclidean and Pearson's similarity.

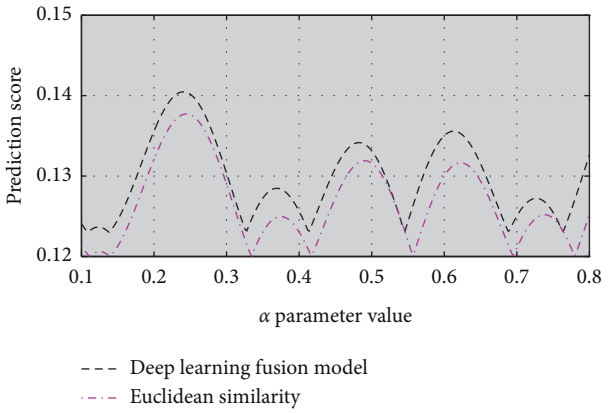

Figure 5: The change curve of the predicted score MAE after case-based reasoning under different α parameters.

After the second stage of refining, the MAE value predicted by the score changes with the α parameter as shown in Figure 5 (in 91% of the training set). This test shows that, in the case of considering certain contextual factors, case reasoning of deep learning fusion model will have a certain effect on improving the prediction score. However, as α increases, the prediction will deviate more and more from the original user's model and become a recommendation that only relies on contextual topics, and the impact of different similarity measurement methods in the collaborative filtering stage on the calculation of the prediction score will also become smaller.

When using Tanimoto's coefficient or log-likelihood ratio similarity, since the input model ignores the preference value, the predicted score and actual score of the test using the data set are both 1 (indicating interest), which means that the MAE is always 0. It is invalid to evaluate the recommendation without preference value in this way, so the accuracy rate and recall rate need to be used to evaluate the recommendation effect using Tanimoto's coefficient and similarity of log-likelihood ratio. When the training set is 91%, the recommendation accuracy rate of the collaborative filtering stage using 4 similarity measures when recommending different numbers of songs is shown in Figure 6.

It can be seen that the accuracy rate increases with the number of recommendations, and there is a period of increasing trend. After too much background music in the recommended video, it can be predicted that the proportion of users who like the video background music begins to decrease. Figure 7 shows the recommended recall rate in the collaborative filtering stage using four similarity measures.

The recall rate basically shows an upward trend with the increase in the number of recommended video background music, but after the number of recommendations reaches a certain value, the increase in the recall rate slows down and even slightly callbacks. This shows that the algorithm can dig out the fact that the user's preference has reached the upper limit. On the whole, the log-likelihood of the data model with no preference value is more suitable for the collaborative filtering stage calculation on the data set than the similarity.

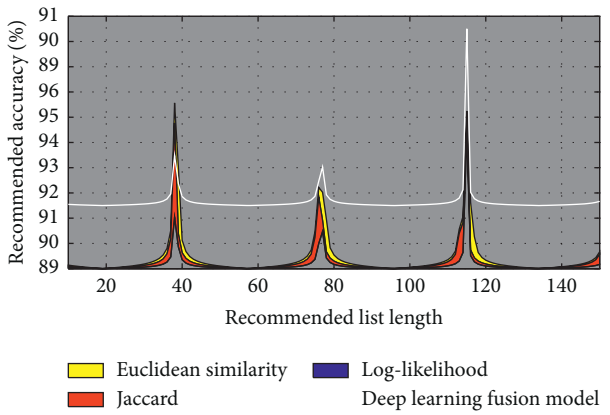

Figure 6: Comparison of the prediction accuracy of Top-N recommends different values of N.

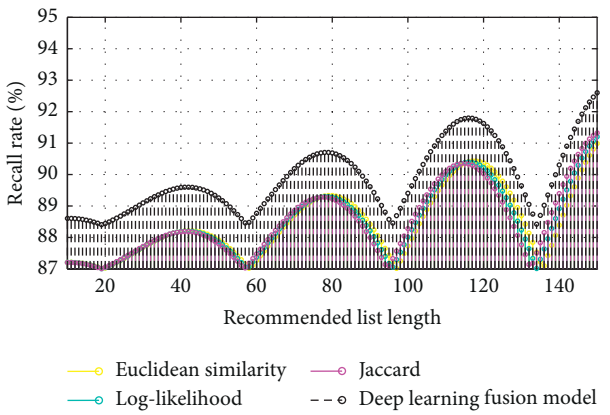

Figure 7: Comparison of predicted recall rate under different recommended N values of Top-N.

After the log-likelihood ratio similarity is used for the first stage of system filtering and screening, after the second stage of refining, the accuracy and recall of the final result vary with the α parameter as shown in Figures 8 and 9 (the recommended number is 150).

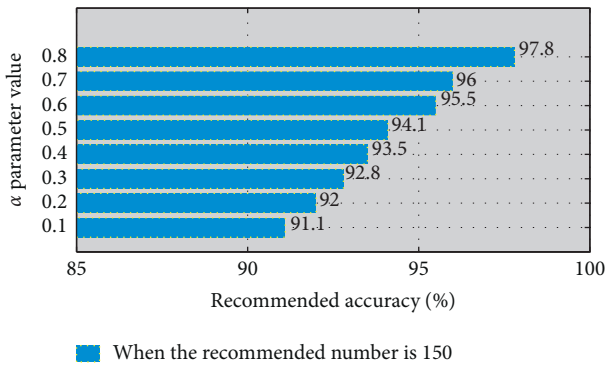

Figure 8: The change of recommended accuracy after case-based reasoning under different α parameters.

It can be seen that, considering certain contextual factors, the recommendation quality has improved after the second-stage case reasoning is refined. However, when the α parameter takes a value of 0.6, the recall rate will drop again, which shows that the user's long-term interest model still accounts for a relatively high degree of importance when making recommendations.

Using the single-machine similarity algorithm on a single server, the average offline similarity calculation takes 23.2 minutes. However, using the distributed cooccurrence matrix method to allocate 4 servers with the same configuration for MapReduce calculation, it only takes 7.1 minutes to complete the similarity calculation on average, which illustrates the effectiveness of using the matrix block method to optimize the efficiency of similarity calculation.

4.2. Evaluation of Test Results. After the Last.fm user data is crawled through the acquisition module, it is preprocessed and written into the scoring database. The information loading time recorded in the user table is shown in Figure 10.

After completing the data preparation, we determine the similarity measurement method and α coefficient according to the above algorithm test, complete the algorithm module configuration, and start the Servlet to wait for user's input information. The user's input interface includes four information input boxes for user's name, song title, artist, and label, and execution buttons. In this experiment, since it is necessary to establish a rating preference model for the existing data set for testing, the user's name here is a necessary condition. The search condition can be song title-artist or tag keywords.

The tracks played for the user are the first songs in the recommended list formed under the search conditions, and in the recommended list, apart from the tracks based on the contextual theme mentioned above, other tracks recommended based on the user's historical evaluation records are also adjusted due to the prediction score adjustment.

Through functional testing, the usability of each module is basically verified. Finally, the system is deployed on the public network server, and each functional module can operate normally. In the Last.fm community, 45 member users from the Asian Music team were randomly invited to

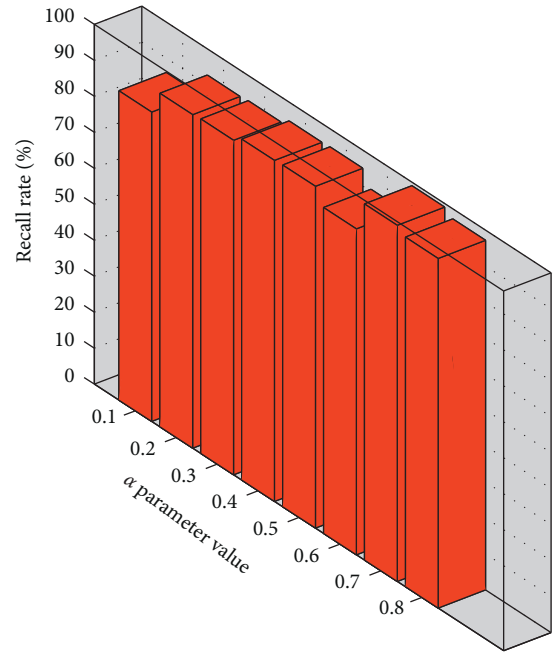

Figure 9: The change of recommended recall rate after case-based reasoning under different α parameters.

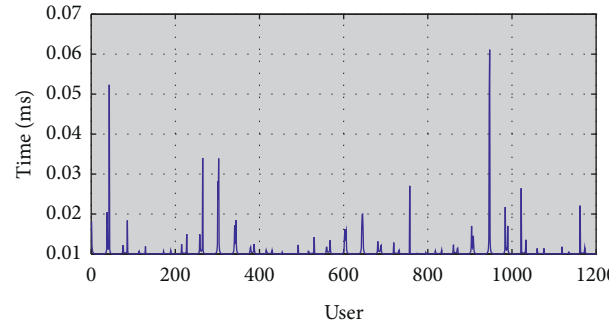

Figure 10: Scoring database user's time.

evaluate the experience of the system. Among them, 41 users had an average rating of more than 4 stars in the feedback of 10 songs in the recommended list, and they also had collection behavior.

In summary, the algorithm designed in this paper can improve the quality of recommendation and can be applied to the radio-type video background music service website to reflect its effectiveness and achieve the expected goal of this subject.

5. Conclusion

In terms of model design, this paper designs a model that combines deep learning model and collaborative filtering. Using the improved stacked denoising autoencoder for structured data processing, user's interest is obtained; using convolutional neural network for unstructured long text mining, the hidden features of video background music are obtained. By introducing the attention mechanism to capture the local key points of the text, the mining effect is

improved, and the interpretability of the model is also strengthened. Based on the idea of probabilistic matrix factorization, the tightly coupled fusion of deep learning and matrix factorization is realized. The model uses a unified objective function to uniformly optimize the two parts of the deep model. The deep model part can provide corrections to the matrix decomposition, and the matrix decomposition part can provide guidance for the feature extraction of the deep learning part, so that the model has a better prediction effect. Aiming at the digital video background music service platform such as video background music network radio station, users need to consider the situation when recommending video background music and develop the method of acquiring context information, the establishment of the recommendation model that adds context information, the context awareness algorithm, and the tradition research work on the fusion method of recommendation algorithms. This breaks through the key technology of short-term user preference discovery, realizes a system that can be applied to the background music recommendation of Internet radio videos, and completes offline comparison tests and online simulation experiments. The results show that the algorithm in this paper achieves a higher recommendation accuracy than the control system: the average absolute error (MAE) of the prediction score can be more than 10% lower than that of the control system, and the recommendation accuracy and recall rate can be improved by more than 20%. Moreover, the system can improve user's satisfaction to a certain extent.

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 no conflicts of interest reported in this paper.

Acknowledgments

This study was supported by Special Topic of Ideological and Political Work in Colleges and Universities in Zhejiang Philosophy and Social Sciences Planning in 2019, Project "Research on Curriculum Ideological and Political Construction Based on Supply-Side Reform: Taking Art Education As an Example" (no. 19GXSZ48YB) and Shaoxing Philosophy and Social Sciences Research 13th Five-Year Plan Key Topics for 2019, Project "Research on Ideological and Political Construction of Art Curriculum in the Perspective of Supply-Side Reform" (no. 135S020).

References

[1] T. F. Tavares and L. Collares, "Ethnic music exploration guided by personalized recommendations: system design and evaluation," *SN Applied Sciences*, vol. 2, no. 4, pp. 1–9, 2020.

[2] M. Jakubec and M. Chmulik, "Automatic music genre recognition for in-car infotainment," *Transportation Research Procedia*, vol. 40, pp. 1364–1371, 2019.

[3] I. Andjelkovic, D. Parra, and J. O'Donovan, "Moodplay: interactive music recommendation based on Artists' mood similarity," *International Journal of Human-Computer Studies*, vol. 121, pp. 142–159, 2019.

[4] S. Roy, M. Biswas, and D. De, "iMusic: a session-sensitive clustered classical music recommender system using contextual representation learning," *Multimedia Tools and Applications*, vol. 79, no. 33-34, pp. 24119–24155, 2020.

[5] M. He, H. Guo, G. Lv et al., "Leveraging proficiency and preference for online Karaoke recommendation," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 273–290, 2020.

[6] A. S. Ulaganathan and S. Ramanna, "Granular methods in automatic music genre classification: a case study," *Journal of Intelligent Information Systems*, vol. 52, no. 1, pp. 85–105, 2019.

[7] S. Visnu Dharsini, B. Balaji, K. S. Kirubha Hari, and M. Elangovan, "Music recommendation system based on facial emotion recognition," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 4, pp. 1662–1665, 2020.

[8] 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.

[9] H. Zamani, M. Schedl, P. Lamere, and C.-W. Chen, "An analysis of approaches taken in the ACM RecSys challenge 2018 for automatic music playlist continuation," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 5, pp. 1–21, 2019.

[10] B. Jia, J. Lv, and D. Liu, "Deep learning-based automatic downbeat tracking: a brief review," *Multimedia Systems*, vol. 25, no. 6, pp. 617–638, 2019.

[11] D. Kang and S. Seo, "Personalized smart home audio system with automatic music selection based on emotion," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3267–3276, 2019.

[12] B. Yi, X. Shen, H. Liu et al., "Deep matrix factorization with implicit feedback embedding for recommendation system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591–4601, 2019.

[13] X. Li, Z. Wang, R. Hu, Q. Zhu, and L. Wang, "Recommendation algorithm based on improved spectral clustering and transfer learning," *Pattern Analysis and Applications*, vol. 22, no. 2, pp. 633–647, 2019.

[14] D. H. Lee and J. W. Beauchamp, "Automatic transcription of solo audio into music notation," *The Journal of the Acoustical Society of America*, vol. 145, no. 3, pp. 1709-1710, 2019.

[15] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–38, 2020.

[16] J. Chen, P. Ying, and M. Zou, "Improving music recommendation by incorporating social influence," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 2667–2687, 2019.

[17] I. Kamehkhosh, G. Bonnin, and D. Jannach, "Effects of recommendations on the playlist creation behavior of users," *User Modeling and User-Adapted Interaction*, vol. 30, no. 2, pp. 285–322, 2020.

[18] Y. He and Q. Zhang, "Automatic generation algorithm analysis of dance movements based on music–action association," *Cluster Computing*, vol. 22, no. 2, pp. 3553–3561, 2019.

[19] J. Lee, W. Seo, J.-H. Park, and D.-W. Kim, "Compact feature subset-based multi-label music categorization for mobile devices," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 4869–4883, 2019.

[20] B. Sharma and Y. Wang, "Automatic evaluation of song intelligibility using singing adapted STOI and vocal-specific features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 319–331, 2019.

[21] Z. Wang, B. Cheng, W. Zhang, and J. Chen, "Q-graphplan: QoS-aware automatic service composition with the extended planning graph," *IEEE Access*, vol. 8, pp. 8314–8323, 2020.

[22] A. Ramchandran and A. K. Sangaiah, "Unsupervised deep learning system for local anomaly event detection in crowded scenes," *Multimedia Tools and Applications*, vol. 79, pp. 35275–35295, 2019.

[23] A. K. Sangaiah, J. S. Ramamoorthi, J. J. P. C. Rodrigues et al., "LACCVoV: linear adaptive congestion control with optimization of data dissemination model in vehicle-to-vehicle communication," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[24] B. A. Harshanand and A. K. Sangaiah, "Comprehensive analysis of deep learning methodology in classification of leukocytes and enhancement using swish activation units," *Mobile Networks and Applications*, vol. 25, no. 6, pp. 2302–2320, 2020.