

Research Article

Personalized Analysis and Recommendation of Aesthetic Evaluation Index of Dance Music Based on Intelligent Algorithm

Jun Geng 

School of Art, Shandong University of Finance and Economics, Jinan 250014, China

Correspondence should be addressed to Jun Geng; 20040293@sdufe.edu.cn

Received 20 August 2021; Accepted 18 September 2021; Published 11 October 2021

Academic Editor: Muhammad Javaid

Copyright © 2021 Jun Geng. 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.

In the era of Industry 4.0 and 5G, various dance music websites provide thousands of dances and songs, which meet people's needs for dance music and bring great convenience to people. However, the rapid development of dance music has caused the overload of dance music information. Faced with a large number of dances and songs, it is difficult for people to quickly find dance music that conforms to their own interests. The emergence of dance music recommendation system can recommend dance music that users may like and help users quickly discover or find their favorite dances and songs. This kind of recommendation service can provide users with a good experience and bring commercial benefits, so the field of dance music recommendation has become the research direction of industry and scholars. According to different groups of individual aesthetic standards of dance music, this paper introduces the idea of relation learning into dance music recommendation system and applies the relation model to dance music recommendation. In the experiment, the accuracy and recall rate are used to verify the effectiveness of the model in the direction of dance music recommendation.

1. Introduction

At present, human society has entered the 5G era and Industry 4.0. With the continuous development of big data, artificial intelligence, and other technologies, the amount of data will expand rapidly, and the phenomenon of data overload will become more significant. The 46th Statistical Report on Internet Development in China released by China Internet Network Information Center shows that the total number of Internet users in China will reach 940 million in 2020. Affected by the novel coronavirus pneumonia epidemic, the user scale of online video, online music, short video, and other applications has increased significantly. Users cannot compare and analyze a lot of information in the face of massive data, and information overload has become a potential challenge today [1]. In order to alleviate the problem of information overload, researchers put forward two solutions: search engine and recommendation system.

Search systems tend to have a clear purpose for users, when users know what they need more accurately, they can easily get relevant content through search engine channels.

However, for some exploratory content, such as watching dance and listening to music, users need to explore according to their interests, so search engines cannot accurately meet users' needs [2]. Recommendation system tends to use data mining and exploration, which is an intelligent system applying machine learning and data mining technology. It uses users' historical scores or project content to recommend related content to users. For users, the personalized content provided by the recommendation system will improve the user experience, thus saving them a lot of time. For service providers, recommendation system can help their customers find interesting products in real time so that more consumers are willing to buy products on the service platform and become loyal customers [3]. In recent years, recommendation system has been widely used in online shopping, news portal, music platform, short video, and other fields.

Dance music is regarded as the most creative work of human beings, which expresses thoughts and emotions through melody and sound. In recent years, dance music streaming media has become the main way for users to

consume dance music and the main source of income in the dance music industry. By 2019, the number of online dance music users in China has reached 608 million, and the dance music income has reached 10.3 billion yuan [4]. Recommendation systems are revolutionizing the dance music industry in many ways. Listeners can improve their taste of dance music by using them to constantly discover new dance music. The challenge of recommendation system is to realize a system that can accurately provide interesting dance music for users, so as to understand users' preference for dance music. For example, Spotify can continue to play songs that it thinks are similar to those in the list after playing all the songs in that list, but the recommended results are based on simple dance music genre labels, which make users feel dull after a while. This requires that the dance music personalized recommendation system should accurately and effectively reflect users' personal preferences and need to be adjusted to achieve personalized recommendation for different users. Personalized recommendation system is more complex than traditional dance music recommendation system, which needs to comprehensively consider various behaviors of users and combine dance music feature recognition and audio processing technology to extract dance music features, so as to realize personalized recommendation of dance music [5].

Dance evaluation theory is based on the philosophical content discrimination of dance criticism objects. Philosophy allows us to understand the artistic form of dance, so that it can be appreciated and practiced, and the human body is a tool to express its artistic form. The definition of "dance" by many dancers and philosophers is not enough to explain and distinguish the necessary conditions of dance from other human nonart forms [6]. In fact, the best understanding of dance should be the concept of the main media of human body and music movement and the secondary media of clothing, scenery, and lighting visual dimension. The "remnants" refer to those ceremonial performances that are still being performed but have lost their original purpose. As long as performers and audiences recognize and accept the way they have changed, such performances can still be appreciated and can still play a role in promoting social cohesion and identity. Lady Nigeria's Girinya dance, originally a war dance, is a case in point. In the aesthetics of Tiv dance, men's dance should be as energetic as Girinya dance, but in the past, dance was about what warriors should behave in battle, and now it is about continuity and renewal. I try to describe and evaluate this aesthetics and its adaptation to the changing culture of Tiv people in a way with new meanings [7]. According to the case study of a primary school music education project in Madeira, Portugal, 30 years ago, we analyzed all children and their activities and made statistics in the form of questionnaires [8]. The results revealed that all respondents showed a strong sense of ownership and leadership. The condition of art critics is to observe artistic experience, which praises them for observing value, but there is not much talk about executive value, so we should discuss whether ideal aesthetic judges should practice and create the artistic form they judge. Researchers put forward that

practicing dance can better observe some aesthetic qualities of dance, and dance training can promote kinesthetic experience. It is precisely because of these experiences that the aesthetic aspects of dance performance can be reflected [9].

With the rapid development of science and technology, robots have gradually entered people's lives. This paper proposes a way for robot to dance with music rhythm, which is a humanoid robot formed by synchronizing music to form dance. Firstly, the audio features are extracted, and then the dance is generated through the action formed by the model. This method forms an artificial creation system, which makes choreographers pay more attention to and exclude dance movements. Through the test on Aldebaran NAO human form and real-life dance experiment, it is evaluated in the aspects of movement, rhythm accuracy, and aesthetics, and the effect is satisfactory and accepted [10]. Finally, the robustness and flexibility of the system enable us to embed it into the artificial creation system in the future work.

2. Individualized Analysis and Recommendation Technology of Dance Music

The recommendation engine uses different algorithms to filter data and recommend the most relevant items to users. It first captures the customer's past behavior, collect personal preferences, calculate the user model on this basis and then combine the user model with the recommendation algorithm model to generate a recommendation algorithm, calculate the scoring level between the user and the target item, recommend the items with higher scores to the target user, and recommend the products that the user may buy. The general model of the recommendation system is shown in Figure 1.

2.1. Analysis of Dance Music Evaluation Index. On the basis of literature research, this study proposes that dance music recommendation system includes 10 dimensions: content, functionality, page design, response speed, perceived usability, perceived usefulness, satisfaction, confidence, perceived experience, and potential risk. The interview results show that due to the upgrade of media equipment and the improvement of network speed, the dimension of "response speed" is of little significance and should be deleted. In addition, some items in the dimensions of "satisfaction" and "potential risk" have semantic intersection with items in other dimensions, which cannot accurately reflect the measurement intention and should be deleted. Considering that the personalized recommendation evaluation index does not point to specific fields, some items are not differentiated and cannot fully summarize the use psychology of dance music users. This study adds the characteristics repeatedly mentioned by many respondents and incorporates them into the index system, such as "surprise degree", "consistency of browsing habits" and "potential risks".

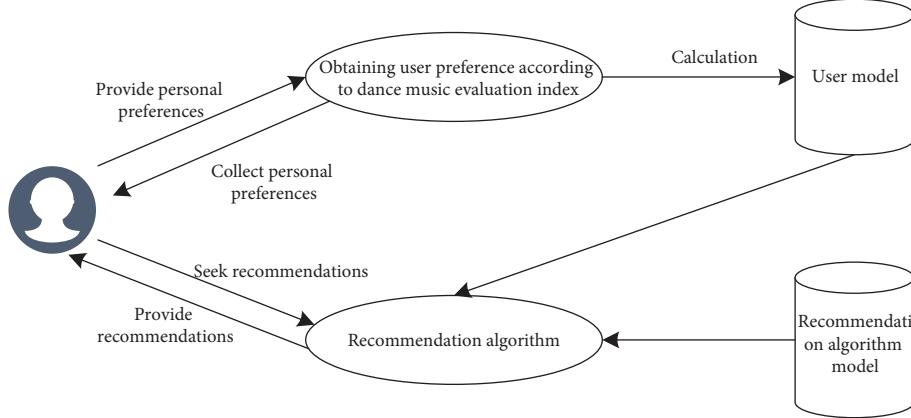


FIGURE 1: Dance music recommendation system model.

TABLE 1: Summary of original evaluation index system of music recommendation system users.

| Second-order index | First-order index | Second-order index | First-order index |
|--------------------|--|-----------------------|---------------------------|
| Content | Accuracy | Perceived ease of use | Convenience of use |
| | Familiarity | | Feedback convenience |
| | Novelty | | Time-consuming |
| | Diversity | Perceived usefulness | Efficiency |
| | Surprise degree | | Preference influence |
| | Sufficiency | Confidence | Quality confidence |
| Functionality | Update of speed | | Recommendation confidence |
| | Requirement considerations | Perceptual experience | Taste confidence |
| | Diagnostic ability of perceptual information | | Dependence on confidence |
| | Preference correction | | Caring for each other |
| Page design | Explanatory power | | Interactivity |
| | Communication ability | | Human touch |
| | Being clear and understandable | Potential risk | |
| | Adequacy of repertoire | | |
| | Information richness | | Risk |
| | Style consistency | | |
| | Consistency of browsing habits | | |

“Surprise” comes from the user’s expectation of dance music recommendation function. Different from other types of recommendation systems, dance music is a perceptual artistic product, and users are more likely to accept the recommended content, thus generating emotional expectations. At present, the personalized recommendation function of most dance music apps in China is on the homepage and occupies an important layout. Users take the consistency of page design and browsing habits as a consideration index, which reflects the needs of users for web page interactive experience in the Internet age. In addition, many interviewees expressed their concerns about the quality of recommendation, including “uneven quality of recommendation”, “frequent recommendation of similar songs that they are not interested in”, and “interest-related but dislike”.

According to the qualitative research results, this paper generated the evaluation scale of dance music recommendation system, which consists of 8 dimensions and 30 items: content, functionality, page design, perceived ease of use, perceived

usefulness, confidence, perceived experience, and potential risk. See Table 1 for measurement indexes and their sources.

2.2. Collaborative Filtering Recommendation.

Collaborative filtering is a method for personalized recommendation according to the ratings and usage behaviors of system users. The idea behind its approach is that if a group of users have similar views on one topic, they may also have similar interests on another topic [11]. Collaborative filtering algorithm does not need to analyze the project content, but mainly pays attention to the user’s scoring data of the project. The system will collect the interaction data between users and projects, establish a scoring matrix, then use the interaction matrix to calculate the similarity between users, and recommend related projects according to similar users. Users can obtain recommendations for items that have not been previously discovered, but which have been positively evaluated by neighboring users. Collaborative

filtering algorithms are mainly divided into two categories: memory-based collaborative filtering and model-based collaborative filtering.

2.2.1. Collaborative Filtering Based on Nearest Neighbors

(1) *Collaborative Filtering Based on Users.* Resnick first proposed a collaborative filtering algorithm based on users in 1994. By using the scoring matrix $R(U, I)$ to model the preferences of users U about goods I , it is considered that users' preferences generally do not change with time and may have similar behavioral preferences for a long time. Therefore, users with similar preferences in user history data are grouped to recommend products to those users.

In the music and dance recommendation system, the historical data used are the operation data such as users listening to songs, whether they like or collect songs, and the scores of dances. For example, both user 1 and user 2 often listen to popular music, and user 3 likes to listen to English music. Through similarity calculation, it can be concluded that user 1 and user 2 are the same kind of users, and user 2 is recommended by using the music listened by user 1. The principle of user-based collaborative filtering algorithm is shown in Figure 2.

In order to make recommendations to users, it is necessary to collect users' historical behavior records and establish a user-item matrix $R_{m \times n}$, where m represents the number of active users and n represents the number of items. By collecting the explicit scores of users on the project, the user's interest in the project is generally indicated by a score level of 0 to 5. The higher the score, the more interested the user is in the project, and the score of 0 indicates that the user dislikes the project at all. In addition, you can use user implicit ratings to represent user ratings. Use the user to listen to a certain music times, or praise, collect, comment and other behaviors, use 1 to indicate that there are related behaviors, and 0 to indicate that there is no browsing behavior. After weighting the above behaviors, the implicit evaluation matrix of the user on the project is obtained [12].

It is difficult to find the similarity degree for such abstract concepts as users, so it is usually to find the similarity degree of users' scoring on specific items and compare the similarity degree of scoring behavior between two users by analyzing the similarity degree of scoring results of two users on the content that they jointly evaluated excessively. In this way, the problem is transformed into a problem of solving the similarity between vectors: R_i and R_j are defined as the content set evaluated by two users, and V_i and V_j are defined as the scoring vectors of the content by two users in R_i and R_j . Generally, the method of calculating the similarity between the two vectors is defined as follows.

Jaccard coefficient method: the similarity is obtained by calculating the ratio of intersection and union of two users' scores in the user history behavior set. The larger the ratio, the higher the similarity. Generally, Jaccard coefficient is suitable for implicit user behavior similarity calculation. The Jaccard coefficient is calculated as shown in

$$\text{Jaccard}(i, j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|}. \quad (1)$$

Cosine similarity method: the cosine value of the angle between two user rating vectors in space is used to express the similarity between them. The closer the cosine value is to 1, the more similar the two vectors are. The calculation process is shown in

$$\cos(i, j) = \frac{\sum_k v_{ik} v_{jk}}{\sqrt{\sum_k v_{ik}^2} \sqrt{\sum_k v_{jk}^2}}. \quad (2)$$

Pearson correlation coefficient method: decentralization of cosine similarity data is an improvement of cosine similarity method. In the recommendation system, there may be missing evaluation of a certain item by users. Pearson correlation coefficient method automatically fills the missing value as 0 and then uses other dimensions to reduce the dimension average of missing vectors to satisfy cosine similarity algorithm. Pearson correlation coefficient is calculated as shown in

$$\text{Pearson}(i, j) = \frac{\sum_k (v_{ik} - \bar{v}_i)(v_{jk} - \bar{v}_j)}{\sqrt{\sum_k (v_{ik} - \bar{v}_i)^2} \sqrt{\sum_k (v_{jk} - \bar{v}_j)^2}}. \quad (3)$$

Finally, according to similar users traversing all positive feedback items, weighted summation is carried out, the interest scores of user I for all items are calculated, and the score prediction is made for each item. The score calculation is shown in

$$P_{i,s} = \frac{\sum_{j \in S(u,k)} \text{sim}(i, j) \times r_{i,s}}{\sum_{j \in S(u,k)} |\text{sim}(i, j)|}, \quad (4)$$

where $S(u, k)$ is a user browsing item set similar to user i , $r_{i,s}$ is user i 's interest in item S , and $\text{sim}(i, j)$ is the similarity between user i and j .

(2) *Collaborative Filtering Based on Project.* Project-based collaborative filtering method mainly focuses on identifying the similarity of projects. However, project-based collaborative filtering looks for users who have used two projects for similar evaluations. If both items Music 1 and Music 2 receive similar ratings throughout the rating data set, it can be assumed that users who like Music 1 may also like Music 2, and vice versa. The principle of project-based collaborative filtering is shown in Figure 3:

Similar to the user-based collaborative filtering method, the item-based collaborative filtering method uses the user's rating of other content to predict the rating of unevaluated content. Similarly, $\omega(i, j)$ needs to be defined between the content that has been evaluated and the content that has not been evaluated, indicating the degree of similarity between R_i and R_j . The formula is shown in

$$\omega(i, j) = \sum_k \frac{v_{i,k} \times v_{j,k}}{\sqrt{\sum_{r_i \in R_i} v_{i,k}^2} \sqrt{\sum_{j \in R_j} v_{j,k}^2}}. \quad (5)$$

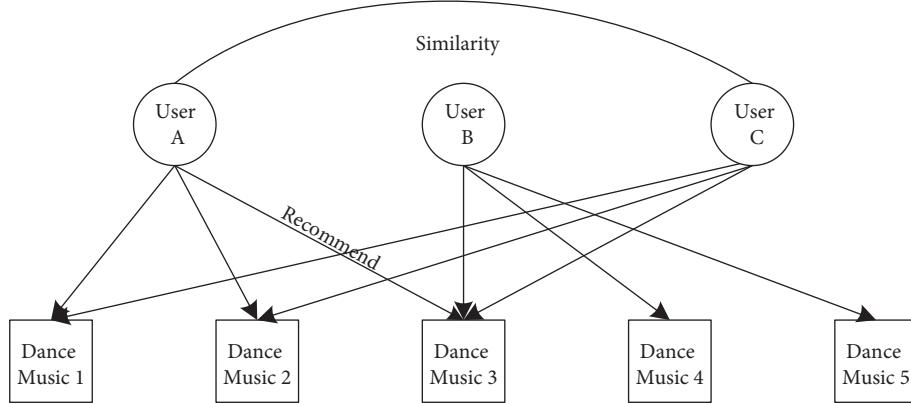


FIGURE 2: Schematic diagram of user-based collaborative filtering.

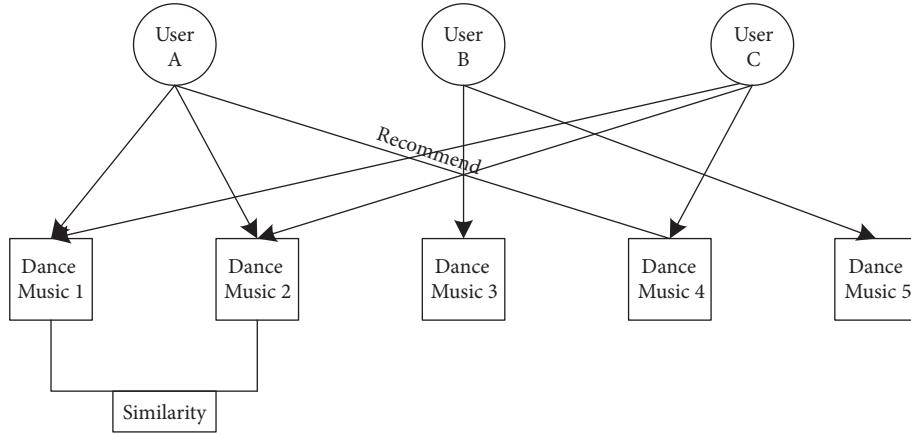


FIGURE 3: Schematic diagram of project-based collaborative filtering.

After that, the unknown scores can be estimated in different ways, such as weighted sum. Here, $C(U, K)$ is a similar set of items in the user evaluation data set, and $r_{u,i}$, i is the interest of user u in item i . The calculation is shown in

$$\tilde{P}_{i,s} = \frac{\sum_{j \in C(u,k)} \omega(i, j) \times r_{u,s}}{\sum_{j \in C(u,k)} |\omega(i, j)|}. \quad (6)$$

Although the user-based collaborative filtering algorithm is logically simple, it may take a lot of time to deal with big data. The project-based collaborative filtering method solves the problem of complex user matrix. By using a significantly smaller number of items, large-scale matrix operations in the user-item scoring matrix can be avoided, and project-based collaborative filtering is more time-cost-effective to implement.

Item-based collaborative filtering depends to a great extent on the number of times songs appear in the data set. Because it is impossible to determine whether songs exist in the user playlist, the item-based collaborative filtering system designates those songs that appear in many playlists in the data set as more valuable data, in this way acting as the basic processing for item-based collaborative filtering. First, match each song in the playlist with a similar song in the song data set and record it; the system records it as “similar

tracks”, and the playlist obtains the most frequently appearing songs from the “similar tracks” matrix, comprehensively considers the user’s preference information for songs, and then recommends music to other users who are similar to the user’s preferences and often listen to them. In the music recommendation system, although the number of songs is less than the number of hundreds of millions of users, it is also very large. Each user playlist may contain more than 50 to 100 songs, so it is much more difficult to find the similarity of songs in the database of tens of millions of songs, so the performance advantage of item-based collaborative filtering is lost [13].

2.2.2. Model-Based Collaborative Filtering. Model-based collaborative filtering is obviously different from the other two collaborative filtering methods, user-based and content-based methods estimate the unknown item scores directly according to the statistical historical data, but model-based methods first use the original data to train and generate a specific model, which can quickly predict user preferences according to the generated model. Therefore, when there are a large number of users and projects, the model-based recommendation algorithm has a high degree of scalability and rapidity. Common models are as follows.

(1) *Bayesian Model.* Bayesian prediction model is a kind of prediction using Bayesian statistics. Bayesian statistics is different from general statistical methods, which does not use model information and data information, but uses prior information. Through the method of empirical analysis, the prediction results of Bayesian prediction model are compared with those of ordinary regression prediction model, and the results show that Bayesian prediction model has obvious advantages.

Bayesian model revolves around giving evidence (E) and obtaining hypothesis (H), which involves two concepts: hypothesis probability $P(H)$ before obtaining evidence and hypothesis probability $P(HE)$ after obtaining evidence. The model is continuously learned by training data, and the data are verified to evaluate the model and make new predictions.

In practical applications, Naive Bayesian classifier and collaborative filtering are generally used to implement recommendation system, which can filter new information and predict whether users need given resources. However, it is almost impossible to obtain a completely independent set of data in practice. Moreover, when the number of items jointly evaluated is small, this direct calculation may distort the obtained probability.

(2) *Matrix Decomposition Model.* Matrix decomposition model is a commonly used technology to build recommendation system. In 2006, Netflix held Netflix Prize Challenge, and matrix decomposition technology was first proposed. In order to decompose the scoring matrix features into low-dimensional spaces, the matrix decomposition technology learns potential space vectors for each user and each item, which are divided into user feature vectors p and item feature vectors q . The specific representation of the scoring matrix is shown in

$$R = p_i \times q_j^T. \quad (7)$$

Here R represents the training score matrix, and i and j represent users and projects, respectively. By learning the minimized objective function (formula (8)) in the known scores, the potential space vectors P and q can be obtained.

$$Y_{i,j} = F^{\text{LFM}}(p_i, q_j) = p_i^T q_j. \quad (8)$$

The algorithms commonly used in matrix factorization model include nonnegative matrix factorization (NMF) algorithm and singular value decomposition (SVD) algorithm. The nonnegative matrix factorization algorithm uses project content to construct the potential space of the project and then uses user information, such as known scores, to learn the potential space of users. So as to overcome the cold start problem in collaborative filtering 28. Singular value decomposition (SVD) algorithm maps users and items to the potential factor space of f -dimension, and the model constructs users and items as interaction problems within the space.

In addition, model-based collaborative filtering algorithms also include graph-based walking method, SimRank algorithm, and clustering-based collaborative algorithm. Collaborative filtering algorithm can recommend various favorite contents to users according to historical data, but

the recommendation results may not be satisfactory when the scoring data are sparse, large-scale matrix operation may be needed when the data volume is huge, and the time consumption will increase linearly with the data scale, so real-time recommendation cannot be achieved.

2.3. *Content-Based Recommendation.* The content-based recommendation algorithm takes into account the items of interest that users have previously shown through ratings and then constructs user profiles that match the user's interest in such items. Once the user clearly shows interest in an item, the algorithm will analyze the characteristics of the item and then recommend similar items to the user. The content descriptor of a project can have many different forms, which can usually be described by tags or project contents: for example, user tags and tags related to project attributes and related types are generally classified by manual or machine learning algorithms. When a user listens to a popular song about campus, the content-based algorithm will recommend another type of campus song to the user.

Early content-based recommendation algorithms mostly deal with text content, so early content-based recommendation systems mainly use text retrieval technology. Vector space model (VSM) can match documents according to keywords and transform song names and singer names into vectors by using word frequency-inverse document frequency (TF-IDF) technology. TF is the number of words appearing, and IDF represents the information of words. If a word appears more frequently in lyrics or song titles, it may play a key role in lyrics. The TF-IDF definition is shown in

$$\text{TF-IDF}(t_k, d_j) = \text{TF}(t_k, d_j) \log \frac{N}{nK}. \quad (9)$$

The TF-IDF algorithm is used to calculate the words with greater weight in the data, and the cosine similarity algorithm mentioned above is used to find other songs similar to this song and recommend them to users.

3. Research on Recommendation Model Based on User Behavior and Music and Dance

Referring to the correlation model of image classification, this paper introduces the model of Relationship-Learning into the field of music recommendation. In this model, deep neural network and automatic encoder are used to combine the characteristics of user behavior and music audio and input them into Music Encoder to realize personalized coding in the coding stage to ensure the personalization of recommendation. By calculating the similarity of user's interest preference and the similarity of song feature, the prediction score of user's music to be selected is obtained after averaging them.

3.1. *Model Introduction.* Through reading the literature, we know that the cold start problem in the recommendation system is mainly divided into two subproblems: complete cold start and incomplete cold start, and there are some

differences in the solutions of these two problems. At the same time, the processing process of image classification direction in small sample data can be used for reference by recommendation system. For example, the classification problem of some pictures that have not been seen at all is Zero-Shot, while the classification problem of some pictures that appear less frequently is Few-Shot. Among these two problems, the Zero-Shot problem can correspond to the complete cold start problem in the recommendation problem, while the Few-Shot problem can correspond to the incomplete cold start problem in the recommendation problem. For the Few-Shot problem and Zero-Shot problem, a similar solution to the cold start problem in the recommendation system is given.

For the cold start problem, one of the existing methods is to use user information and article information. If effective information can be collected, certain clustering operations can be carried out on this basis, and other similar users' favorite music can be used to recommend the user. In addition, the structure of Auto Encoder can be used to extract features. At this time, for the cold-start user, the Auto Encoder encodes first and then decodes, and the finally decoded value includes the value inferred from the user characteristics [14].

In this paper, we pay more attention to the problem of incomplete cold start when designing the model and use the limited scoring information as efficiently as possible to analyze the scoring habits of users more accurately. At the same time, it draws lessons from the idea of relationship learning in the field of computer vision, hoping that the recommendation system can learn how to compare similarity in a more accurate way. On this basis, we investigate the application of metric learning in recommendation system and find that although these methods have the idea of making recommendation system learn distance information to infer similarity, the distance that defines similarity is low-order linear similarity. Although this method can give a certain distance description, it is difficult to fully capture and utilize the high-order features extracted by neural network. This paper hopes to calculate the similarity between pictures in relation learning and apply it to the recommendation system to calculate the similarity between people, music and music, user and music.

3.2. Metric-Learning Model. In 2018, Sung and Flood proposed a relation network model based on the metric-learning method to solve the classification problem of a small number of labeled sample data in the field of computer vision in deep learning, as shown in Figure 4. The author proposes a relational network, which compares the input images in the test set with a small number of known labeled sample images and calculates the correlation scores, so as to classify them. The relation network consists of two modules: an embedded module and a relational module. Firstly, the embedding module extracts the relevant feature information, and then the relational module compares these embedding and obtains the relation score to determine whether they belong to the same class.

Each block has 64 3×3 convolution kernels, a batch specification layer, and ReLU nonlinear function. In addition, convolution blocks 1 and 3 have a 2×2 maximum pooling layer. The author uses four convolution blocks to extract rich feature information between samples with fewer parameters. The network structure is shown in Figure 5.

In the relation module, you need to combine features first. The feature combination S recombines the feature information of the samples in the training set and the sample information in the test set so that the relational encoder G can learn better from the combined features. First, the feature mappings of the same class are summarized, as shown in

$$f_\varphi(L_i) = \sum_{j=1}^K L_{i,j}, \quad (10)$$

where $L_{i,j}$ ($i = 1, 2, \dots, j = 1, 2, \dots, K$) denotes the training set features extracted from the embedded module and $f_\varphi(L_i)$ denotes the mapping of feature information. The mapping of combined features can be obtained by summing the mapped feature sets. The formula is shown in (11):

$$S_i = f_\varphi(L_i) + f_\varphi(Q_n). \quad (11)$$

S_i represents the feature formed by the feature mapping of the training data set and the recombination of the test samples, and $f_\varphi(Q_n)$ represents the feature from the test samples. The model does not use conventional Euclidean distance or cosine distance; instead, we use nonlinear metric learning. In this part, two convolution blocks and two fully connected layers are used to compare the two samples. Each convolution block has 64 3×3 convolution kernels, a batch specification layer, a ReLU nonlinear activation layer and a 2×2 maximum pooling layer. Finally, through two fully connected layers, the similarity relation score $\in [0, 1]$ between the test samples and the training samples is finally obtained by using the activation functions ReLU and sigmoid. The relation score is shown in

$$\text{Relation score} = g_\phi(S(f_\varphi(L_{i,j}), f_\varphi(Q_n))). \quad (12)$$

The detailed structure of the module is shown in Figure 6.

3.3. Construction of Relation Model. The basic idea of this model is determined as a hybrid recommendation model, learning the similarity of user ratings by using the idea of Relationship-Learning and then estimating the rating value through collaborative filtering. The model structure is shown in Figure 7.

This model mainly deals with two problems in recommendation system. Firstly, timeliness: in the current era of information explosion, a large amount of data is uploaded to the Internet every day. It is difficult to obtain enough rating data at the beginning of these newly uploaded data, which can be used in the recommendation system for target users. Music recommendation system often encounters some new songs, which need to be recommended to users for audition.

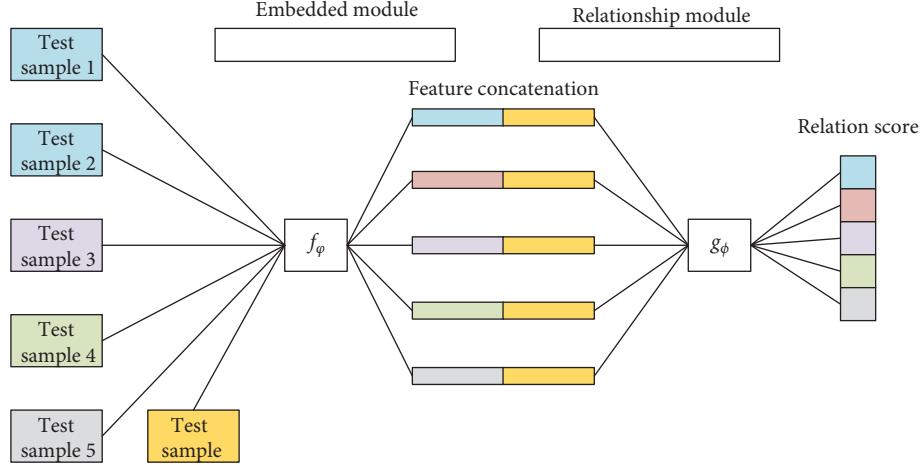


FIGURE 4: Relation network model.

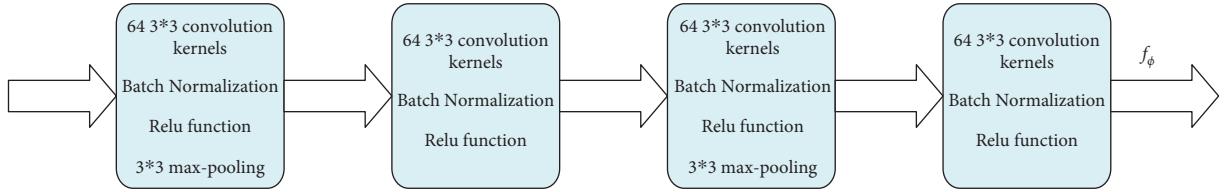


FIGURE 5: Embedding module.

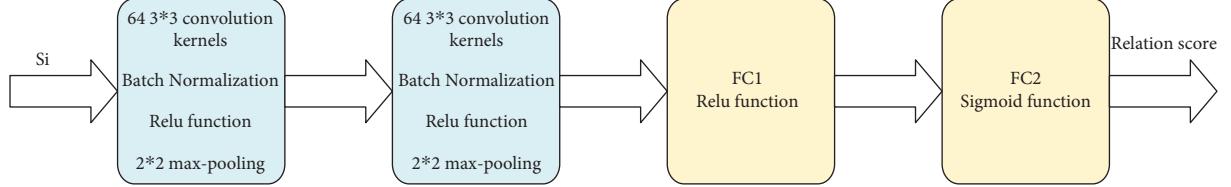


FIGURE 6: Relation module.

Therefore, through the cold start mechanism of the model, the problem of new content recommendation can be solved, and the recommendation system can cope with the problems brought by timeliness and new content.

Secondly, personalization: when users evaluate songs, even if two people give favorable comments on the same song, their motives may be different; for example, one person likes the singer so they like this song, and the other person likes the song style and likes this song. Therefore, the model carries out more personalized processing for song recommendation according to music characteristics and user characteristics.

3.3.1. Enter Data. The input of the model mainly consists of two parts: one part is the feature information, which includes the personal feature information of the target user, the feature information of the scored n songs, and the feature information of the target songs. The other part is the scoring information, which contains the scoring values of the target users for n songs that have been scored. In the third chapter, the related features have been extracted. Formally, the related features are expressed as follows:

Users: $U = \{field_1^u, field_2^u, \dots, field_n^u\}$ represents the user's gender, age, and other related characteristic information. Music characteristics: $M = \{field_1^M, field_2^M, \dots, field_t^M\}$ represents the relevant characteristic information of music. Dance features: $D = \{field_1^D, field_2^D, \dots, field_j^D\}$.

Historical interaction data of user U : $H = \{M_1, M_2, \dots, M_k\}$, user U 's comprehensive user score for music M is $Score_i$, and user U 's comprehensive user score for dance D is $Score_j$.

3.3.2. Embedding. According to the previous feature extraction work, several user and audio features can be obtained. Firstly, the features are encoded through linear layer or embedding layer to obtain a series of vector representations of features; then, according to these vectors, the user and music are encoded. The user below represents several characteristics of the target user, which are represented by vectors. Taking it as input, the encoding vector of user features is obtained through the User Encoder. For user coding, because only text type features such as gender and age are used, Auto Encoder model can be directly used to complete feature extraction. In order to be similar to Music

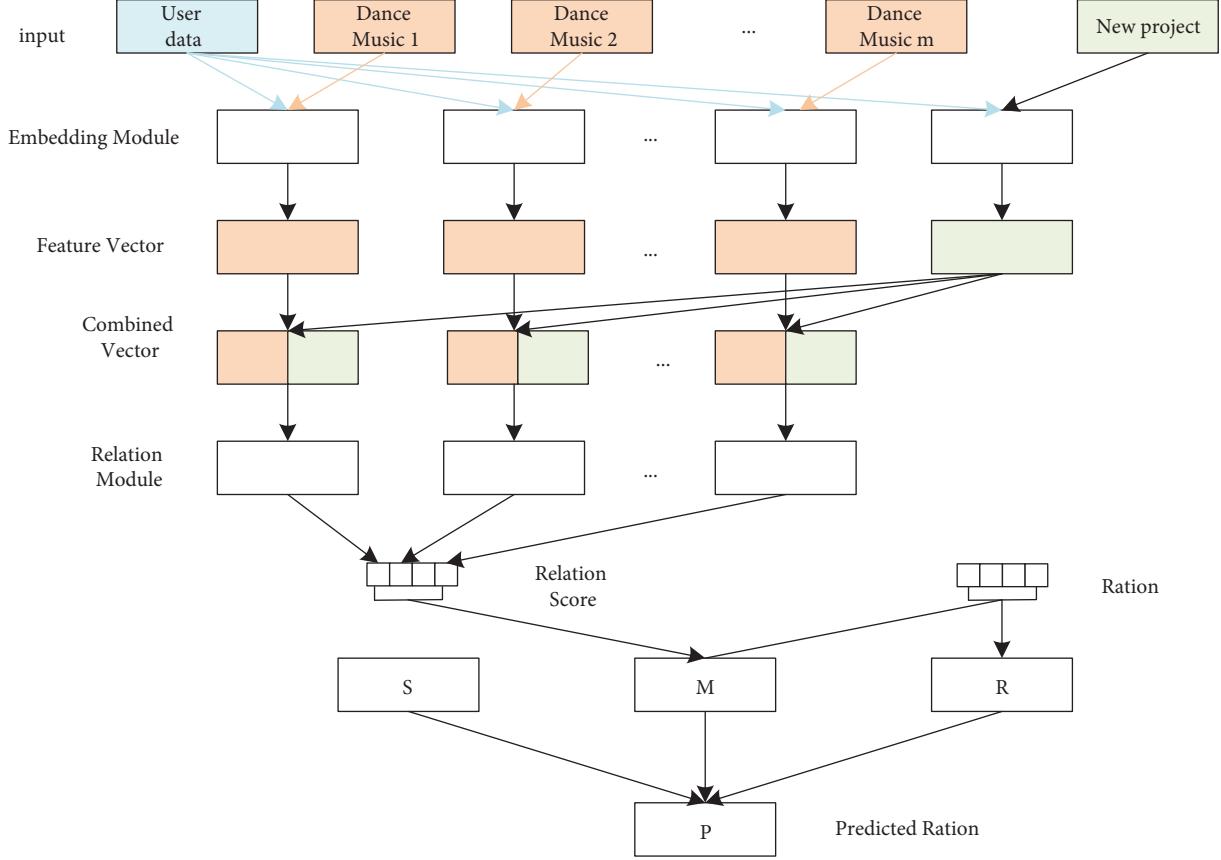


FIGURE 7: Model structure.

Encoder feature structure, AutoInt model l is used to extract user features here. The encoding structure of the user part is shown in Figure 8.

Next, for some music evaluated by users, because they know their characteristics and scores, they can combine their characteristics with user's feature vectors and input them into Music Encoder for coding, respectively. This process can combine users' personalized information for music, so that the same music can be coded differently by Music Encoder for different users. For unevaluated music, although there is only the eigenvalue of music, it can still obtain its eigenvector through Music Encoder like the music that has been evaluated, which is the coding layer, Embedding Layer of music. The music feature coding structure is shown in Figure 9.

As can be seen from Figure 9, the main structure of encoder is multihead attention module, and its essence is to perform multiple self-attention calculations to form multiple subspaces, so that the model can learn different features of information from different angles and finally merge them. Take the Music Encoder as an example, and its structure is shown in Figure 10.

In the multihead attention, the feature matrix $F = \{f_1, f_2, \dots, f_t\}$ needs to be transformed linearly to compare the dimension of the feature vector, that is QW_i^Q . Perform k self-attention calculations as shown in

$$h^* = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{d_k}\right)V. \quad (13)$$

Splice and linearly map the result h^* using

$$h = \text{concat}(h_1^*, h_2^*, \dots, h_n^*) \otimes W^o. \quad (14)$$

The number of music selections has also been adjusted to some extent. When entering, you need to have a series of known characteristics of the score and its final score. However, there will be many historical records of a person. And the quantity is different, and the model structure does not support the dynamic adjustment of the input quantity, so this paper chooses the sampling method. During each training, five historical evaluated songs are selected from the historical music of user interaction for input. After several rounds of training, all the evaluated songs can be basically used in the training. Although this method is inefficient in using data sets, it can greatly reduce the training time cost and the complexity of the model. Therefore, after trade-off, we chose the method of randomly selecting five data.

3.3.3. Interest Matching Module. The information about users and music in the data set is processed, and the obtained high-order nonlinear eigenvector is taken as the output. In

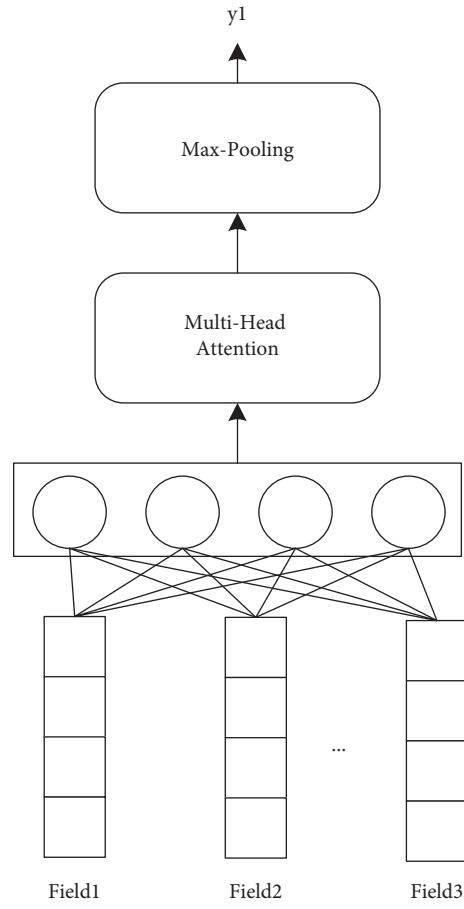


FIGURE 8: User encoder model structure.

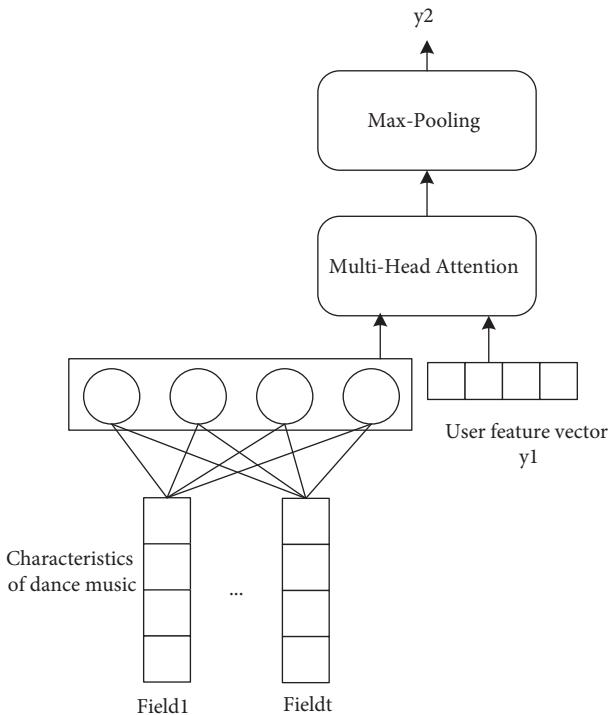


FIGURE 9: Music encoder model structure.

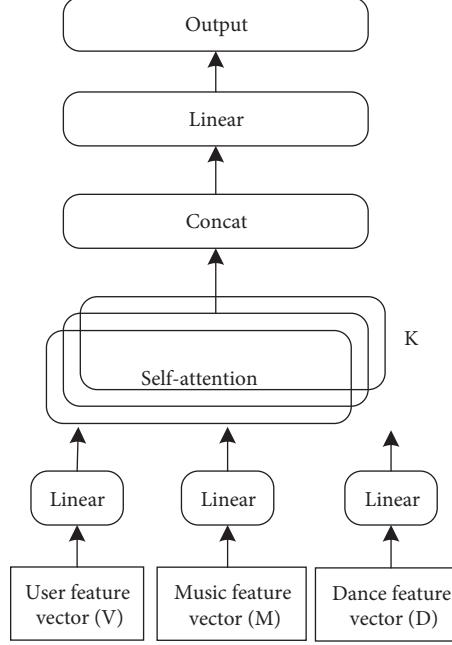


FIGURE 10: Multihead attention module.

the model of this section, we hope to give the similarity of users about different music scores through feature vectors.

Firstly, the characteristics of the scored songs and the characteristics of the target unscored songs are combined by the user. Then the similarity $g\varphi$ between the two songs is given by comparing the similarity module and relation module, its size is limited between 0 and 1 by normalization function, and finally, a k-dimensional numerical vector relation score is obtained. The similarity $g\varphi$ model is shown in

$$g\varphi(q_1, q_2) = \text{Sigmoid}(q_1^T W q_2). \quad (15)$$

The Interest Matching Relation module constructed in this paper is not complex, and the parameter to be trained is only a matrix W . In the experiment, we also use a deeper structure to replace this structure. It is found that this will not only increase the training time and parameter scale, but also reduce the accuracy. Therefore, we finally choose the value between [0, 1] obtained by sigmoid after we get the inner product of two vectors about W as the similarity. According to the similarity degree, the related features of the known graded music are constructed to predict the features of the music. The method is weighted average, and the weight of each item is the score multiplied by the similarity degree. The more similar the music and the higher the score, the more obvious the influence on the prediction results. Finally, an estimated value of the eigenvector is obtained.

Finally, there is an output layer, which synthesizes the two scoring results given before and gives the final conclusion. In the first part of the conclusion Y from the previous user coding, music-to-be-predicted estimation features, and music-to-be-predicted coding through a linear layer, this part reflects the idea of collaborative filtering based on similarity and score to get the target score. Another part of the result x is obtained by using a Max Pooling layer

to select the value with the greatest correlation. The strategy embodied in this part is that when the songs to be predicted have a high similarity with the previous historical songs, it can be considered that users are likely to have a certain interest in the songs to be evaluated, so the results have a large similarity value at this time. Finally, the two scores are averaged to get the final output. This is a value between [0, 1] to indicate the degree of interest that users may be interested in.

4. Implementation of Personalized Recommendation

Traditional models do not use the information of users who interact with them in the coding process of known music. This leads to the lack of user personality in the recommended content. Even for two different users, the coding of the same content is the same, which cannot deal with the personalized problem well. In addition, in the actual interactive process, content and users have different interactive behaviors, and different interactive behaviors often represent different preferences of users. However, the previous models have not been distinguished to this extent, resulting in different information brought by different interactive behaviors not being used. For example, for praise and collection, collection means that users are more interested than praise, because collection means that users are likely to listen to this song repeatedly later, while the probability of listening to praise will be smaller later.

In this paper, firstly, different behavior scores of users are distinguished, and the scores are divided into 1–5 points according to the listening times, 6 points for collection behavior and 4 points for praise behavior. In addition, the multihead attention mechanism is introduced. When

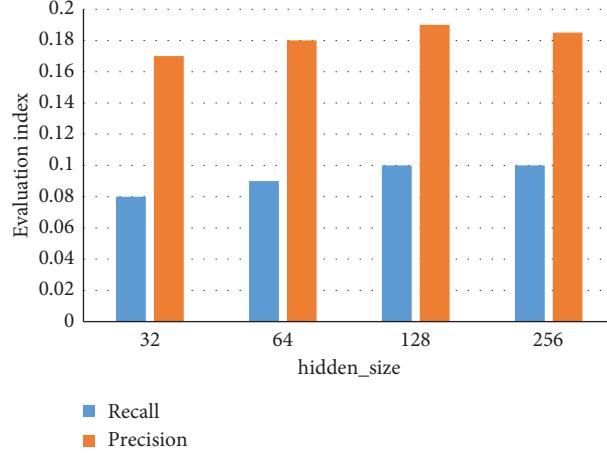


FIGURE 11: Impact of hidden size on evaluation indicators.

encoding music features in Music Encoder, the user features and music features are combined, and the user personalized music coding is realized in the encoding layer, which ensures that even if different users operate the same music, the final recommendation results are still different because of different personal characteristics, thus realizing the personalized music recommendation.

4.1. Model Training and Experimental Results. Before training the model, it is necessary to divide the positive and negative classes and determine the training set and test set used by the model. For music recommendation, label 0 means dislike and label 1 means like, which is defined as a binary classification problem. According to the scoring processing of user behavior characteristics in Chapter 3, music with a score higher than 5 is recorded as a positive class, that is, if the user listens to music many times, or music with a little praise or collection behavior is recorded as a positive class. Music with a score below 2 is recorded as a negative example, indicating that although the user has some interaction, they have not shown interest. Negative cases with the same number of positive classes are randomly selected from music with scores lower than 2 to ensure that the distribution is as balanced as possible. After dividing positive and negative classes, the first 80% of the data set is used as the training set and the last 20% as the test set.

4.2. Super Parameter Adjustment. The hyperparameter adjustment process aims to optimize the performance of the model. First, select the super parameters that can be adjusted. Then, determine whether they will be fixed or variable, and if the parameters are changeable, set them to different values to determine in which range they will change.

Hidden_size: the original model is set to 50 and features are represented using the model by setting the number of hidden layers. The hidden layer is set to 32, 64, 18, and 256 in the implementation process, and the best parameters are determined by experiments in TP100 data set in Figure 11.

According to Figure 11, when the hidden layer is set to 32, the model is in an underfitting state and cannot be characterized. With the increase of the hidden layer, the accuracy rate and recall rate are continuously improved, but when the hidden layer is set to 256, both indicators decrease slightly. Therefore, in the subsequent experiment, the hidden_size is set to 128.

Learning_rate: it depends on the Adam optimizer in the model. The learning rate of each parameter is adapted by making smaller updates to frequent parameters and larger updates to infrequent parameters. The learning rate of the original model is set to 0.001. In this experiment, the learning rates are set to 0.001, 0.01, 0.1, and 0.5, respectively. According to Figure 12, when the learning rates are 0.001 and 0.01, the accuracy is better. In the subsequent experiment, the learning_rate is set to 0.01.

Batch_size: when smaller batches are used, a period of training is more detailed, which usually leads to a decrease in the number of convergent iterations, but the training time is long. On the other hand, the more the batches, the slower the convergence speed, which reduces the risk of overfitting and reduces the training time. The effect of batch_size on accuracy is shown in Figure 13.

Try to use the batch size training model from 100 to 1000 in the whole process. As can be seen from Figure 13 and Table 2, using a smaller batch size, the training time is longer. Using a large batch size (when it exceeds 700), a memory error was encountered.

Therefore, in the follow-up experiment, the batch size is set to 600 for model training, which not only meets the training time, but also does not overflow the memory.

4.3. Experimental Results. The whole training process Loss-Epochs is shown in Figure 14. During each training, five songs with historical evaluation are selected from the historical music interacted by users for input. After several rounds of training, all the evaluated songs can basically be used in training, and the training loss is basically stable after 1024 iterations.

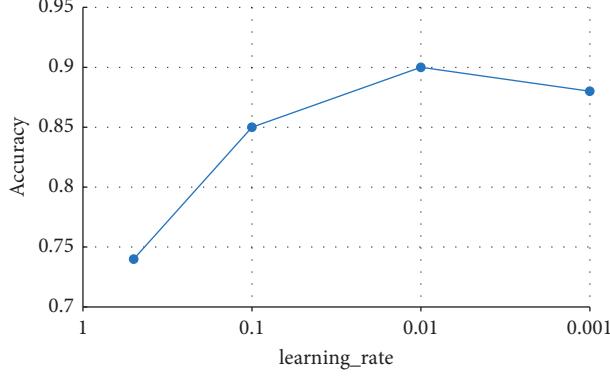


FIGURE 12: Impact of learning rate on accuracy.

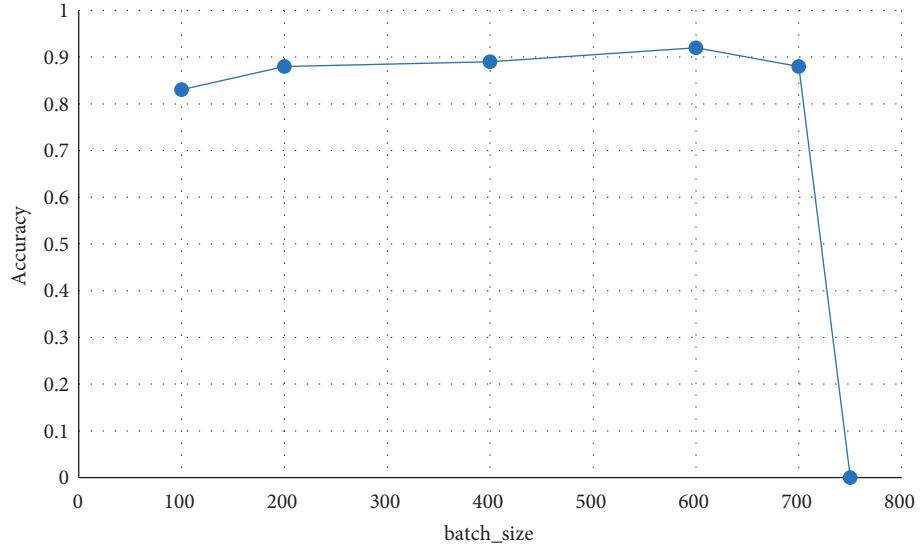


FIGURE 13: Effect of batch_size on accuracy.

TABLE 2: Average running time per Epoch.

| batch_size | 100 | 200 | 400 | 600 | 700 |
|-------------|-----|-----|-----|-----|-----|
| Runtime (s) | 67 | 39 | 20 | 9 | 8 |

In order to verify the effectiveness of the model, this paper implements Neural Network Based Collaborative Filtering (NCF) model and SVD model by Python, which are verified in TP100 and TP500 data sets, respectively. The accuracy and recall rate are used to evaluate the accuracy of different recommendation lengths. The accuracy results of different recommendation lengths are shown in Figure 15.

It can be seen from Figure 15 that the accuracy of the deep learning recommendation model based on Relationship-Learning is obviously higher than that of the traditional SVD recommendation algorithm in two data sets, about 3% higher than that of the collaborative filtering algorithm

based on neural network in TP100 data set, and similar in TP500 data set with lower sparsity, both of which are higher than SVD algorithm.

As can be seen from Figure 16, this model has obvious advantages in both data sets. With the increasing number of recommendations, the recall rate gradually increases, which is about 10% higher than the traditional SVD algorithm. In TP500 data sets with dense data, the recall rate reaches about 0.3. To sum up, the deep learning recommendation model based on relationship learning implemented in this paper can accurately predict users' music preferences and finally can accurately recommend music lists for users.

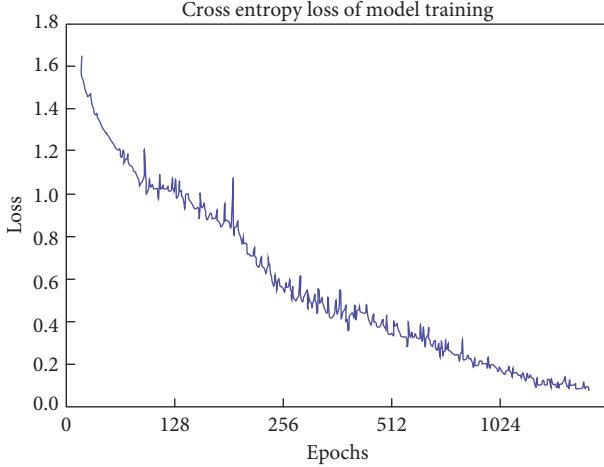
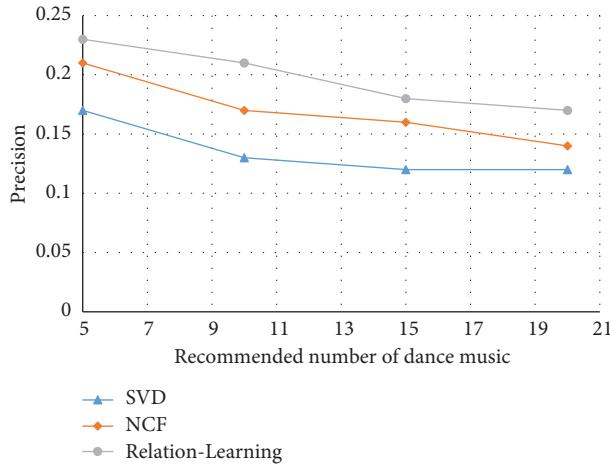
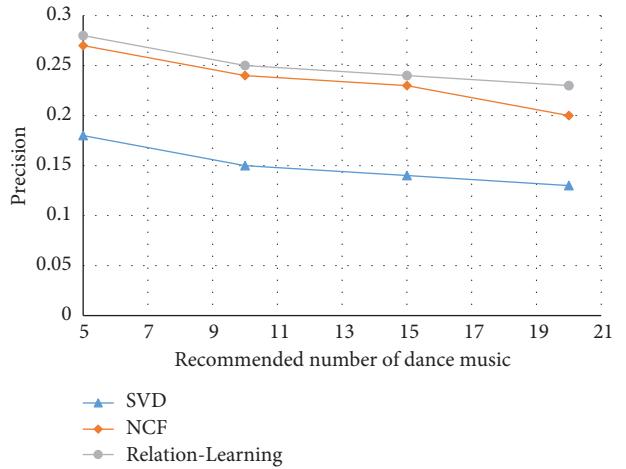


FIGURE 14: Model training loss.

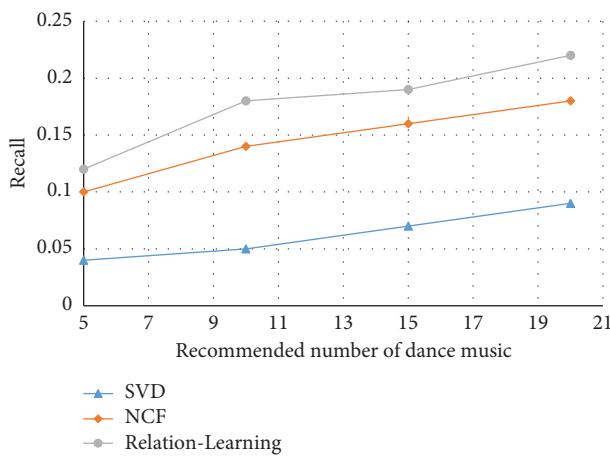


(a)

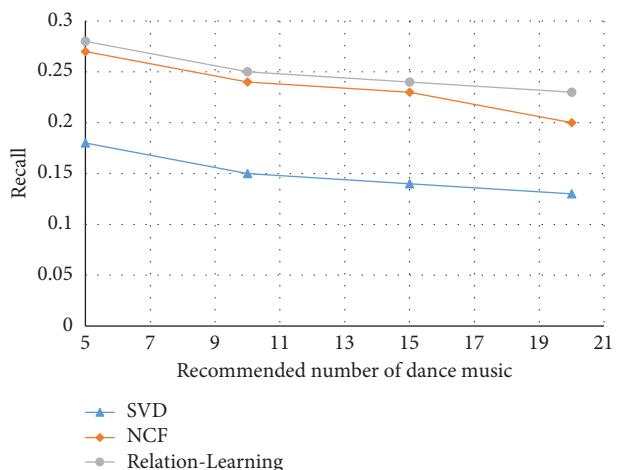


(b)

FIGURE 15: Accuracy of two data sets. (a) TP100 data set. (b) TP500 data set.



(a)



(b)

FIGURE 16: Recall rate of two data sets. (a) TP100 data set. (b) TP500 data set.

5. Conclusion

Faced with a large number of dances and songs, it is difficult for people to quickly find dance music that meets their interests. The emergence of dance music recommendation system can recommend dance music that users may like and help users quickly discover or find their favorite dances and songs. This kind of recommendation service can provide users with a good experience and bring commercial benefits, so the field of dance music recommendation has become the research direction of industry and scholars. In this paper, relation learning is introduced into dance music recommendation system, and the relation model is applied to dance music recommendation. In the experiment, the accuracy and recall rate are used to verify the effectiveness of the model in the direction of dance music recommendation.

Data Availability

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

Conflicts of Interest

The author declares that no conflicts of interest regarding this work.

References

- [1] Y. Han, Z. Han, J. Wu et al., “Artificial intelligence recommendation system of cancer rehabilitation scheme based on IoT technology,” *IEEE Access*, vol. 8, pp. 44924–44935, 2020.
- [2] A. Rw, M. A. Xiao, and J. A. Chi, “Heterogeneous information network-based music recommendation system in mobile networks,” *Computer Communications*, vol. 150, pp. 429–437, 2020.
- [3] X. Wen, “Using deep learning approach and IoT architecture to build the intelligent music recommendation system,” *Soft Computing*, vol. 25, no. 1, pp. 1–10, 2020.
- [4] H.-G. Kim, G. Y. Kim, and J. Y. Kim, “Music recommendation system using human activity recognition from accelerometer data,” *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 349–358, 2019.
- [5] M. S. Fathollahi and F. Razzazi, “Music similarity measurement and recommendation system using convolutional neural networks,” *International Journal of Multimedia Information Retrieval*, vol. 5, pp. 1–11, 2021.
- [6] G. Sun, Y. Wong, and Z. Cheng, “DeepDance: music-to-dance motion choreography with adversarial learning,” *IEEE Transactions on Multimedia*, vol. 99, p. 1, 2020.
- [7] J. M. Agaku, “The Girinya dance theatre of the Tiv people of Nigeria: an aesthetic evaluation,” *International Journal of Intangible Heritage*, vol. 3, no. 6, pp. 29–41, 2008.
- [8] G. Mota and L. Abreu, “Thirty years of music and drama education in the Madeira Island: facing future challenges,” *International Journal of Music Education*, vol. 32, no. 3, pp. 360–374, 2014.
- [9] B. Montero, “Practice makes perfect: the effect of dance training on the aesthetic judge,” *Phenomenology and the Cognitive Sciences*, vol. 11, no. 1, pp. 59–68, 2012.
- [10] A. Manfrè, I. Infantino, F. Vella, and S. Gaglio, “An automatic system for humanoid dance creation,” *Biologically Inspired Cognitive Architectures*, vol. 15, pp. 1–9, 2016.
- [11] D. Aledo, B. Carrion Schafer, and F. Moreno, “VHDL vs. SystemC: design of highly parameterizable artificial neural networks,” *IEICE - Transactions on Info and Systems*, vol. E102.D, no. 3, pp. 512–521, 2019.
- [12] B. Sheng and G. Sun, “Matrix factorization recommendation algorithm based on multiple social relationships,” *Mathematical Problems in Engineering*, vol. 2021, Article ID 6610645, 8 pages, 2021.
- [13] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, “A trust-based collaborative filtering algorithm for E-commerce recommendation system,” *Journal of ambient intelligence and humanized computing*, vol. 10, no. 8, pp. 3023–3034, 2019.
- [14] A. Dm, A. Xl, and A. Qd, “Humming-query and reinforcement-learning based modeling approach for personalized music recommendation,” *Procedia Computer Science*, vol. 176, pp. 2154–2163, 2020.