Hindawi
Security and Communication Networks
Volume 2023, Article ID 9756418, 1 page
https://doi.org/10.1155/2023/9756418

Retraction

Retracted: Hybrid Music Recommendation Algorithm Based on Music Gene and Improved Knowledge Graph

Security and Communication Networks

Received 11 July 2023; Accepted 11 July 2023; Published 12 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] T. Zhang and S. Liu, "Hybrid Music Recommendation Algorithm Based on Music Gene and Improved Knowledge Graph," *Security and Communication Networks*, vol. 2022, Article ID 5889724, 11 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5889724, 11 pages
https://doi.org/10.1155/2022/5889724

Research Article

Hybrid Music Recommendation Algorithm Based on Music Gene and Improved Knowledge Graph

Tingting Zhang and Shengnan Liu

College of Music, Shijiazhuang University, Hebei 050035, China

Correspondence should be addressed to Tingting Zhang; 1101838@sjzc.edu.cn

Received 16 February 2022; Revised 10 March 2022; Accepted 21 March 2022; Published 9 April 2022

Academic Editor: Chin-Ling Chen

Combining music as a specific recommendation object, a hybrid recommendation algorithm based on music genes and improved knowledge graph is proposed for the traditional single recommendation algorithm that cannot effectively solve the accuracy problem in music recommendation. The algorithm first gives the recommendation pattern of music genes and gets the relevant recommendation results through the genetic preference analysis. After that, the algorithm in this paper utilizes item and user label information and knowledge graphs from two different domains to enrich and mine the potential information of users and items. In addition, deep learning method is applied to extract low-dimensional, abstract deep semantic features of users and items, based on which, score prediction is performed. The mixed-mode based recommendation addresses the drawbacks of these two recommendations and can adopt different weighting strategies in different situations. The advantages of music gene and knowledge graph-based recommendation algorithms are combined via this method. The experimental results indicate that the algorithm in this paper outperforms other existing recommendation algorithms.

1. Introduction

With the rapid development of mobile Internet, smart terminals, and Internet of Things technologies, a wide variety of information such as text, audio, video, images, and social networks are growing in an explosive situation on the Internet, enriching people's daily life and learning and working content [1, 2]. The rapid development of information technology and Internet technology has generated a huge amount of information, which has greatly not only enriched people's personal needs but also brought about the problem of information overload. Therefore, recommendation systems have emerged, aiming to better meet users' personalized needs and solve the information overload problem. In this era of information overload, the Internet has become the most important part of people's lives. However, a challenge to filter out the interesting contents from the huge amount of Internet information takes up. Recommender systems, which personalize and recommend objects to meet users' needs based on their interests and other characteristics, have been widely used in e-commerce, information portals, social networks, mobile location services, multimedia entertainment, and other fields [3, 4].

Music is an important entertainment element in people's life, and with the development of information technology, music resources are growing in a huge amount [5]. Personalized recommendation, as an effective solution to the information overload problem, has received more extensive attention in the music field [6] and has been widely used. Almost all music platforms currently provide personalized music recommendation services, such as Spotify, Pandara, QQ Music, KuWo Music, and WangYi Cloud Music. Many platforms have achieved a good reputation for the accuracy of recommended music. If only the user and the recommended item are considered, it will affect the performance of the recommendation system to some extent. A very important part of personalized music recommendation system is to consider the user's contextual information at that time and make reasonable use of the user's context for personalized recommendation [7–9]. However, the current music

recommendation algorithm is more of a single recommendation algorithm.

Due to the characteristics of music with rich variety, large quantity, short listening time as well as coherence and sequence, the traditional single recommendation algorithm does not address the accuracy of music recommendation in a targeted way. For example, Last.fm is recommended by collaborative filtering, and Pandora is recommended by content similarity [10]. Wang proposed a hybrid recommendation algorithm based on reinforcement learning. The algorithm uses deep learning and weighting matrices to extract song features, considering the problem of changing listener preferences [11]. Feng et al. incorporated attention mechanisms into user history behavior and constructed a hybrid model to finally achieve music recommendation [12], and in both the literature [13, 14], music genes are mixed with recommendation algorithms to achieve music recommendation.

More and more data in the Internet can be accessed sensitively, and multisource heterogeneous auxiliary information including tags, images, and text can be used to enrich the description of users' personalized needs and items and enhance the mining capability of recommendation algorithms. This effectively alleviates problems, such as data sparsity and cold start, and improves the accuracy, diversity, and interpretability of recommendation algorithms. Tags are keywords with no hierarchical structure and employed to describe information, which can be used to describe the semantics of items [15]. Many web-based information service systems, such as product recommendation systems Taobao, Weibo, and Douban, allow users to add tags to items to facilitate information search and item recommendation. Tags serve as a bridge between users and items, expressing the characteristics of items and users' preferences for them. Knowledge graph [16] is a graph-based data structure whose nodes represent entities that exist in the real world and edges represent the relationships between entities. Researchers have constructed a series of knowledge graphs covering different domains, which contain rich information about entities and relationships, extending the hidden relationships that exist between users and items. Knowledge graph can efficiently compute the relationships between entities and provides an effective solution for improving the accuracy and interpretability of recommendation models.

The problem of low accuracy of recommendation algorithms due to existing recommendation systems with problems such as sparse data and cold start is addressed in this paper. In addition, for the problem that the scope of music gene recommendation application is relatively smaller, this paper combines music gene, tagging, knowledge graph, and deep learning, and proposes a hybrid recommendation algorithm based on music gene and improved knowledge graph. The algorithm mixes music gene-based recommendation with improved knowledge graph algorithm by weighting. When the user's evaluation information is relatively small, the weight of music gene-based recommendation is added. When the user's evaluation information is more, the weight of the improved knowledge graph-based recommendation algorithm is increased. This allows deep data mining based on music content features and user preferences, song tags, to improve personalized recommendation efficiency and provide a reference method for optimizing music recommendation systems.

Section 2 in this review is an introduction to related work. Section 3 is on the music gene-based recommendation algorithm. Section 4 is a recommendation algorithm based on improved knowledge graphs. Section 5 is the hybrid recommendation algorithm. Section 6 is the conclusion and Section 7 is the conclusion.

2. Related Jobs

2.1. Tag-Based Recommendation. Tag-based recommendation algorithms are also developing rapidly along with the evolution of tagging technology. Usually, tag-based recommendation systems make recommendations by analyzing the information of tags marked on the items by users. Some researchers use the weight vectors of different tags as input to make recommendations by determining the relevant tags of items and the tag preferences of users. Tag2word models further discover the semantics of tag information to improve recommendation methods. Deep learning is widely adopted in tag-based recommender systems. Some researchers first avail tags to represent user features, and then utilize deep neural network models to extract deep features from the latent space of tags layer by layer and apply the extracted features for collaborative user-based filtering recommendations. Deep-semantic similarity-based personalized recommendation (DSPR) exploits deep semantic similarity-based neural networks to map user features and item features into a deep feature space.

2.2. Knowledge Graph-Based Recommendations. As an effective auxiliary information source in a recommendation system, the knowledge graph contains a large amount of relevant information about the items in the recommendation system and rich implicit semantic associations among the items. RippleNet builds a knowledge graph centered on the items that users interact with, and user interests are propagated outward and decayed layer by layer on the knowledge graph. Knowledge graph convolutional network (KGCN) automatically captures the higher-order structure and semantic information of the knowledge graph using graph attention network and learns the potential long-range interest of users, which makes the recommendation with better interpretability.

In recent years, the recommendation method combining knowledge graph and deep learning has become a development trend, which obtains the low-dimensional vectors of entities and relationships in the knowledge graph through knowledge graph representation learning and uses deep neural networks to abstract the implicit features of items and users to improve the recommendation performance. Deep knowledge-aware network (DKN) extracts the relevant background knowledge of news headlines to construct the knowledge graph, and exploits the TranR method to realize the knowledge. Recurrent knowledge graph embedding

(RKGE) is based on deep neural networks and automatically mines the connection patterns between entities in the knowledge graph to learn the relationship between users and items. Knowledge graph enhanced neural collaborative recommendation (K-NCR) is a neural collaborative recommendation method based on deep neural networks for knowledge graph enhancement, which automatically mines and extends the user's potential interests and connection patterns between entities in the knowledge graph.

3. Gene-Based Recommendation Model

3.1. Gene Structure. Genes are originally biological terms for the basic units of heredity that carry genetic information and control the expression of biological traits. Music includes the fundamental elements of melody, beat, and sound quality. The key to distinguishing one piece of music from another is based on the melody and the beat of the music. The concept of musical genes is also meant to represent the various essential information that controls the auditory effects of music.

Most of the current music sharing systems do not have precise tags about the music, thus users are unable to detect this music through precise music text information. It leads to no small waste of resources, and then the necessity to use music genes to serve users. Generally, if a user likes a certain type of music, it must be because the music has a certain musical characteristic that attracts the user. For example, users' preference for the music of violin playing has a certain relationship with the melodious tone of the violin. Music is rich in content, and in addition to additional attributes (such as composer, era, mood, and other social attributes), music itself exists as a combination of various musical elements (melodies, instruments, and other internal musical attributes). Users can explicitly state the name of the music they like and the type of music they like, but they cannot explicitly state the specific model of the music they like.

If we summarize all the attributes of music, we can basically get the following general structure chart of music genes in Figure 1.

For a piece of music, certain characteristics are fixed genes that exist with the music internally, such as melody, tempo, and other characteristics. These are markers that users can employ to perceive words and emotions, so this category of features can be defined as internal genes of music. As a product born in a society, different music has different social attributes, which are again attributed to social genes. These include the only constant deterministic attributes such as the name of the music, the lyricist, the composer, the name of the singer, the gender of the singer, the language, the region, the genre, etc., and the indeterminate attributes such as emotion, suitability for the situation, etc., which are artificially defined.

Whether users are first exposed to internal genetics (hearing the song first) or social genetics (learning something about the song first) for music genetics, users like a piece of music more in favor of the internal genetics of the music. But the reasons for liking may vary. In most cases, the user simply likes the music, which is a recognition of the music's internal genes. It is also possible that a piece of music that most people do not accept may be changed from not very acceptable to acceptable after repeated exposure because of the user's love for the singer or composer of the music, which is a recognition of the music's social genes.

In the external genetic part of a piece of music, there are some genes that are fixed and will not change, such as the name of the music, the name of the artist, and the gender of the artist. However, there are other genes, such as the emotion and style, which can be different for different listeners. A simple musical genetic structure consists of the following parts:

(1) The name of the music: The name that the author or the producer of the song has drawn up for the song, which is an important marker for listeners when searching for and listening to a song.

(2) Language of the music: The language availed to sing that music, such as Mandarin, English, Cantonese.

(3) The geographical region of the music: the country or region where the artist who sings or plays the music is located, for example, listeners are often classified by Europe, America, Japan, and Korea, the mainland, Hong Kong, and Taiwan.

(4) The emotion of the music that is warm or sad. Even the same piece of music produces different feelings in different people's ears, and listeners can get this free gene by analyzing the labels users put on the music.

(5) The style of music: Works written by certain composers in a certain region of the world or at about the same time often have a similar style, but individual composers using the same musical language can also form individual expressions. Common classifications of musical styles include pop, rock, jazz, punk, etc.

(6) Scenes of different music for different scenes age listening, such as sports time suitable for listening to the passionate music, rest of time suitable for listening to soft and relaxing music.

3.2. Genetic Preference Analysis. In the music recommendation system of this review, three genetic preferences are selected for analysis, namely, geographical preference, emotional preference, and style preference.

3.2.1. Area Preference. By observing listeners' listening behaviors to songs, we can know that different listeners like different music from different regions, such as some listeners prefer European and American music, while some listeners prefer domestic music. For each region, the regional preference of users is calculated separately.

$$AP_{i,j} = \frac{\mathrm{SA}(i, j)}{\mathrm{SC}(i)}, \tag{1}$$

where $\mathrm{AP}_{i,j}$ describes the preference of listener i for songs belonging to region j. $\mathrm{SA}(i, j)$ indicates the number of songs

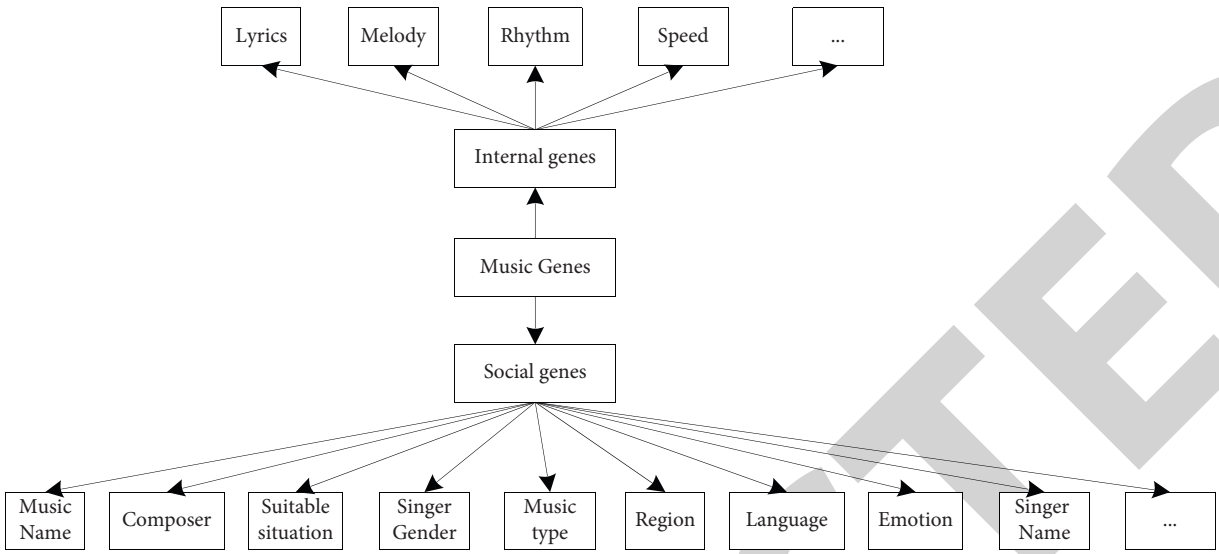

Figure 1: General structure of music gene.

belonging to locale j among all songs listened to by the listener. $\mathrm{SC}(i)$ represents all songs listened to by listener i.

3.2.2. Emotional Preference. Different users have different preferences for music emotions, with some users preferring fresher and warmer ones and others leaning toward slightly sad ones. For each type of music emotion, the user's preference for each emotion is calculated:

$$\mathrm{EP}_{i,j} = \frac{\mathrm{SE}(i,j)}{\mathrm{SC}(i)}. \tag{2}$$

$\mathrm{EP}_{i,j}$ describes how much listener i likes the songs belonging to emotion j. $\mathrm{SC}(i)$ represents all songs listened to by listener i.

3.2.3. Style Preference Degree. Different users have different preferences for music styles, some users like rock punk, others prefer jazz and classical. For each style of music, the user's preference for each style was calculated:

$$\mathrm{SP}_{i,j} = \frac{\mathrm{SS}(i,j)}{\mathrm{SC}(i)}, \tag{3}$$

where $\mathrm{SP}_{i,j}$ indicates the preference of listener i for songs belonging to style j. $\mathrm{SS}(i,j)$ denotes the number of songs belonging to style j in the total songs listened to by listeners. $\mathrm{SC}(i)$ represents all songs listened to by listener i. Finally, the preference of each music attribute above is combined to obtain the genetic preference of the user:

$$P_{i,j} = a^{*}\mathrm{AP}_{i,j} + b^{*}\mathrm{EP}_{i,j} + c^{*}\mathrm{SP}_{i,j}. \tag{4}$$

a, b, and c represent the specific gravity coefficients of the audience to the three attributes of area, emotion and style respectively. The values of a, b and c are set to 0.25, 0.35, and 0.4 respectively, and the sum of the three is 1.

4. Recommendation Algorithm Based on Improved Knowledge Graph

The detailed structure, based on knowledge graph and label-aware recommendation algorithm design is shown in Figure 2, including knowledge graph embedding, user-item modeling, and rating prediction 3parts. The algorithm takes user, item, and label information as input, and obtains feature representations of users and items through the knowledge graph convolutional network to finally predict users' ratings of items.

4.1. Knowledge Graph Embedding. The Knowledge Graph G usually stores entities and their relationships in a triad of "entity-relationship-entity" (b, r, n). The entities and their relations are stored in a triad, where $b \in E, r \in R, n \in E$. E and R are the sets of entities and relations in the knowledge graph, respectively.

4.1.1. Project Entity Embedding. The project embedding reflects the features of project entities. First, DBpedia is applied as a project-oriented knowledge graph to construct a project-centric knowledge graph, and then KGCN is employed to learn the representation of the knowledge graph to obtain the project entity representation vector. KGCN is a graph attention network applied to the knowledge graph, which can be used to capture the topology and the entity information in the knowledge graph. The core idea is to compute the features of a given entity in the knowledge graph by biasedly aggregating and incorporating the information of its neighboring nodes into the features of that entity. The model is a multilayer structure, where the low-order features of an entity can be obtained at the lower level and the higher-order information of an entity can be mined at the higher level.

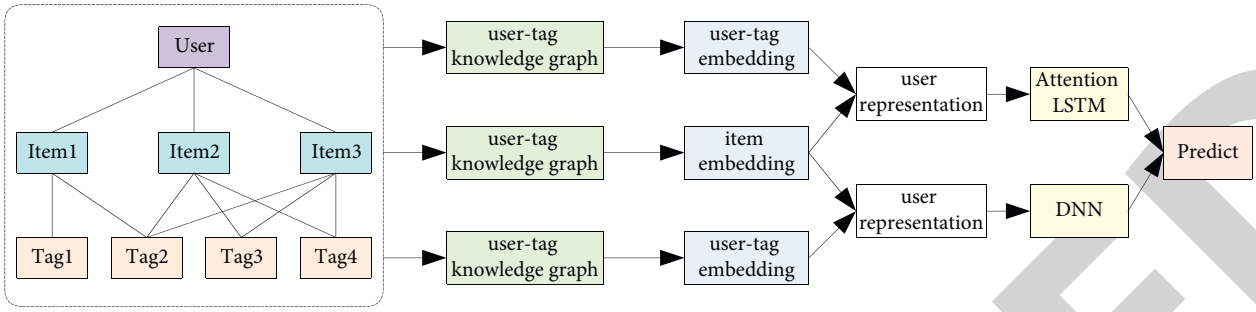

Figure 2: Framework diagram of improved knowledge graph recommendation algorithm.

For users $p \in P$ and items $q \in Q$, where P and Q are the set of users and the set of items, respectively. $T(q)$ denotes the set of entities directly connected to q, and $r_{x,y}$ denotes the relationship connecting entity e_x and entity e_x. The correlation between users and relations is calculated using the vector inner product, which is called the user-relationship score:

$$\pi^p_{r_{x,y}} = p \cdot r_{x,y}, \tag{5}$$

where $p \in \mathbb{R}^d$ and $r_{x,y} \in \mathbb{R}^d$ are the feature representations of the user p and the relation $r_{x,y}$, respectively. d is the feature vector dimension. The user relationship score $\pi^p_{r_{x,y}}$ portrays how important the relationship $r_{x,y}$ is to the user p. For example, a user may have a greater interest in the same type of item. Therefore, a higher attention needs to be given when the relationship $r_{x,y}$ is an item type.

Then the nearest neighbor topology of item q is modeled and the linear combination of domain nodes of item q is calculated:

$$q^p_{T(q)} = \sum_{e \in T(q)} \tilde{\pi}^p_{r_{q,e}} e^0, \tag{6}$$

where $e^0 \in \mathbb{R}^d$ is the initial representation of entity e. $\tilde{\pi}^p_{r_{q,e}}$ is the normalized user-relational score.

$$\tilde{\pi}^p_{r_{q,e}} = \frac{\exp\left(\pi^p_{r_{q,e}}\right)}{\sum_{e \in T(q)} \exp\left(\pi^p_{r_{q,e}}\right)}. \tag{7}$$

In computing the neighborhood representation of an item, the normalized user-relational score reflects the weight of user preferences, and the neighborhood of the item is weighted and aggregated according to the user preference weights.

In a real knowledge graph, the size of neighbors $T(q)$ of different project entities q may vary. To maintain computational efficiency, a fixed number of sets of neighborhoods from the neighbors need to be sampled for each entity. The neighborhood representation of the project entity q is denoted as $q^1_{S(q)}$:

$$S(q) \triangleq \{e | e \sim T(q)\}, \tag{8}$$

$|S(q)| = K$ is the number of neighbor samples.

The initial representation q^0 of the project entity and its neighborhood representation $q^1_{S(q)}$ are aggregated to obtain the project entity representation q_{gaa} that incorporates the project entity neighborhood information:

$$q_{gaa} = \sigma\left(M_{gaa} \cdot \left(q^0 + q^1_{S(q)}\right) + h_{gaa}\right), \tag{9}$$

where $M_{gaa} \in \mathbb{R}^{d \times 2d}$ and $h_{gaa} \in \mathbb{R}^d$ are the weights and biases, respectively. σ is the activation function.

With a single-layer KGCN, the representation of an entity will depend on itself and its neighbors, and call q_{gaa} the first-order representation of item q, denoted as q^1. To explore the potential interest of users in a deeper and reasonable way and to explore the higher-order features of entities, the KGCN is extended from one layer to multiple layers. The initial representation of each item entity (i.e., the zero-order entity representation) is propagated outward and aggregated with the representations of its neighboring entities, and then the first-order item entity representation is obtained. The above process is then repeated, that is, the first-order representation is further aggregated to obtain the second-order representation. In general, the L-order representation of an entity is the aggregation of the entity itself with its neighboring entities in the L-hop range. The L-order representation of the item can be taken as the final item entity representation q_e.

First, the knowledge graph is constructed with the item as the center. Second, the neighborhood of the item is sampled and the user-relational score $\pi^p_{r_{x,y}}$, in the graph3 e0i is the initial representation of the entity. Then the first-order representation of the entity e^0_x is calculated by (6). Finally, the first-order representation is extended to the L level, and the L-order representation of the entity e^L_x can be obtained and used as the final representation of the entity.

4.1.2. Label Embedding. The label information is enhanced by using ConceptNet, a word-oriented knowledge graph, to extend the semantic information of labels, aggregate label features with item or user features, and mine the deeper potential features of user and item-related label attributes. The entities in the tags are first identified, disambiguated using entity linking techniques, linked with entities in the ConceptNet knowledge graph, and their associated triples are extracted. Then similar project entity embedding method is exploited to construct the knowledge graph-centered on the entities in the labels, and KGCN method is applied to obtain the representation of the project label q_n and the representation of the user label p_n.

4.2. Project and User Modeling

4.2.1. Project Modeling. Aggregate the project entity embedding q_e with the project label embedding q_n to obtain the final project representation q:

$$q = q_e \oplus q_n, \tag{10}$$

where $\oplus$ denotes a tandem operation between two vectors. The obtained item characteristics can better reveal the similarity between items.

After obtaining the item embedding, a deep neural network is used to learn the latent features of the item, taking the original item representation $q^{(1)}$ as input and outputting the hidden representation of the item $\widetilde{X}_1$ through an implicit layer:

$$\widetilde{X}_1 = a\left(f\left(q^{(1)}\right)\right) = \mathrm{ReLU}\left(M_1 \cdot q^{(1)} + h_1\right), \tag{11}$$

where M_1 is the weight matrix connecting the input layer to the first implicit layer. h_1 is the bias, and a is the ReLU activation function. In a similar way, the output of the y th layer can be obtained:

$$\widetilde{X} = a\left(f\left(q^{(y-1)}\right)\right) = \mathrm{ReLU}\left(M_p \cdot q^{(y-1)} + h_p\right), \tag{12}$$

where $q^{(y-1)}$ is the output of the $(y-1)$ layer and $\widetilde{X}$ is the potential vector representation of the item.

4.2.2. User Modeling. The user features are aggregated by the item entity embedding q_e and the user label embedding p_n that the user has interacted with according to the temporal organization to obtain the final user representation p:

$$p = q_e \oplus p_n. \tag{13}$$

User characteristics capture both dynamic user preferences for item entities and user preferences for labeled attributes, enabling a better representation of potential user characteristics.

In order to better explore the potential interests of users in the long and short term, a neural network based on the attention LSTM structure is designed, which contains two LSTM layers and an attention layer. The overall structure of this model is shown in Figure 3.

LSTM is a powerful model for learning features from sequence data. Compared with ordinary neural networks, this model performs stepwise analysis of sequences and can model long-range dependencies more effectively by adding hidden layer units that preserve long-term states and storing information from all previous steps in the hidden layer.

The attention mechanism can be used to learn how much the user's historical data contribute to the prediction goal, and thus can effectively reflect the user's behavioral preferences. The attention mechanism calculates the weights of the items that the user has interacted with, and the dynamic preference representation of the user can be obtained by weighting and summing the interaction history representations with these weights. Each item of the vector in the user interaction history $\left\{q_{e,1}, q_{e,2}, \ldots, q_{e,Z}\right\}$ is assigned weights

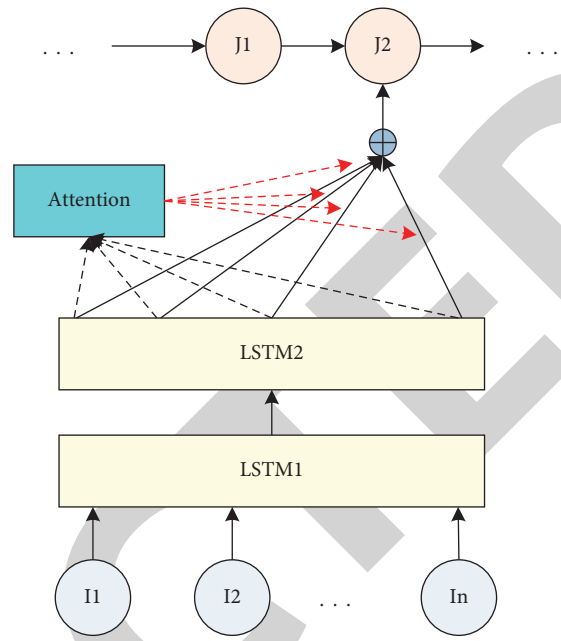

Figure 3: The structure of attention LSTM model.

and weighted equally to obtain the dynamic preference representation of the user.

$$\widetilde{P} = \sum_{x=1}^{Z} g_x q_{e,x}, \tag{14}$$

where, g_x is the attention weight, which is calculated as follows:

$$\begin{aligned} a_x &= \tanh\left(M_{gnn} \cdot q_{e,x} + h_{gnn}\right), \\ g_x &= \frac{\exp\left(a_x\right)}{\sum_{y=1}^{Z} \exp\left(a_x\right)}, \end{aligned} \tag{15}$$

where, M_{gnn} and h_{gnn} are the weights and biases of the attention mechanism.

4.3. Scoring Prediction. The learned latent features of users and items are used to predict the corresponding ratings. In the matrix decomposition method, the inner product of the user potential vector p_x and the item potential vector q_y can be approximated by r_{xy}:

$$r_{xy} \approx p_x \cdot q_y. \tag{16}$$

Therefore, the inner product of two potential feature vectors learned from the hybrid deep structure can be used to obtain the prediction of users' ratings of items:

$$\hat{r}_{px} = \mathrm{pred}\left(\widetilde{P}, \widetilde{X}\right) = \widetilde{P} \cdot \widetilde{X}. \tag{17}$$

The loss function is the squared error between the true and predicted scores and is the minimization objective of the algorithm, defined as follows:

$$L(\theta) = \sum_{r_{px} \in R} \left(\hat{r}_{px} - r_{px}\right) + \lambda \|\Theta\|^2, \tag{18}$$

where, r_{px} is the actual score. The hyperparameter λ prevents overfitting, which is achieved by controlling the strength of L_2 regularization, and Θ is the set of parameters to be trained. The stochastic gradient descent method is used to minimize the loss function. This algorithm is the basic optimization algorithm in optimization theory, which first finds the direction of the fastest descent by finding the partial derivatives of the parameters, and then continuously optimizes the parameters by iterative method.

5. Mix Recommendation

The main hybrid methods used in current recommender systems can be divided into the following types:

(1) Weighted. By weighting the results of multiple recommendation algorithms to form a final list of recommendations.

(2) Switching.

(3) Mixed.

(4) Waterfall (cascade). By using the latter recommendation algorithm to improve the results obtained by the previous recommendation algorithm. This is a two-step process, the first step uses a recommendation algorithm, the results obtained in this step are relatively coarse, on top of this result, the next step uses another recommendation algorithm to make more accurate recommendations on the results generated by the previous step.

In the hybrid recommendation model of this study, a weighted hybrid recommendation approach is used, that is, collaborative filtering-based and music gene-based, each generates a recommendation result, which is then weighted and mixed to obtain the final recommendation list.

The overall flow of hybrid recommendation is shown in Figure 4.

6. Experiments and Analysis of Results

6.1. Experimental Data Set and Evaluation Index. Dataset1: Last.fm is the world's largest social networking site for music. Users from all over the world listen to songs on this site and communicate with other users. Last.fm also records the songs that each user has listened to. In this experiment, we downloaded the Last.fm dataset and selected a part of it as the experimental data. The selected dataset consists of 2,315 of 29,413 listeners, 29,413 songs, and 16,384 tags.

Dataset2: QQ Music is the most used music software in China, and the software commonly used rate is up to 54.3%. The music dataset2 was obtained by extracting the software data using a crawler tool. The data set contained 3,522 one listener, 35,343 songs, and 10,429 tags.

In the experiments, for each listener, all the music in the dataset is divided into three components A, B, and C, where A is used as the training set and B is used as the test set, with A and B accounting for 8 : 2 of the music listened to by the listener, respectively. c denotes the set of music that the listener has not been exposed to.

In this review, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are employed as evaluation criteria, abbreviated as MAE and RMSE, respectively. MAE is used to measure the quality of recommendation results, and RMSE is availed to measure the variance of recommendation results. The lower the value of MAE, the higher the quality of the recommendation. RMSE is the sum of squares of the differences between the predicted and true values, and can be applied to evaluate the volatility of the predicted results.

6.2. Experimental Results. To evaluate the performance of the algorithm when different network types and different network depths are chosen, a bottom-up approach is used to create different models for comparison. In the algorithm, λ is set as the 0.01, learning rate as the 0.01, and embedding dimension as 100 dimensions. The optimal parameters of the three-layer neural network is determined by experiments on the dataset that are set.

The model1 uses three-layer DNN and single-layer LSTM to learn the latent features of items and users, respectively.

The model2 uses three-layer DNN and two-layer LSTM to learn the latent features of items and users, respectively.

The model learns the latent features of items and users using a three-layer DNN and a two-layer LSTM incorporating an attention mechanism, respectively.

The impact of different structures on the algorithm is shown in Table 1.

From the experimental results, the MAE and RMSE values of the model2 using DNN and two-layer LSTM are 1 lower than those of the models, indicating that the two-layer LSTM structure will have a positive impact on the algorithm performance. This is because the lower layer neural network captures the surface features of the input data, while the deeper layer neural network can extract the higher-level semantic abstraction and obtain the implicit feature representation of the data. Based on the two-layer LSTM structure, the model with the attention mechanism is better 3 than the model2 performance, indicating that the attention mechanism can learn the user's preference weights for items from the user's interaction data and reflect the user's interest, thus improving the recommendation performance. The proposed two-layer LSTM structure with attention mechanism can learn the complex nonlinear relationships in the data through the deep interaction of potential features, and shows better recommendation performance.

Knowledge graph representation learning is to embed the knowledge graph in a low-dimensional space, and the recommendation effect achieved by different embedding dimensions will be different. For the dimensionality of item entity and label embedding, the experiments are conducted in 50–250 dimensions based on the model3, and the experimental results are shown in Figure 5.

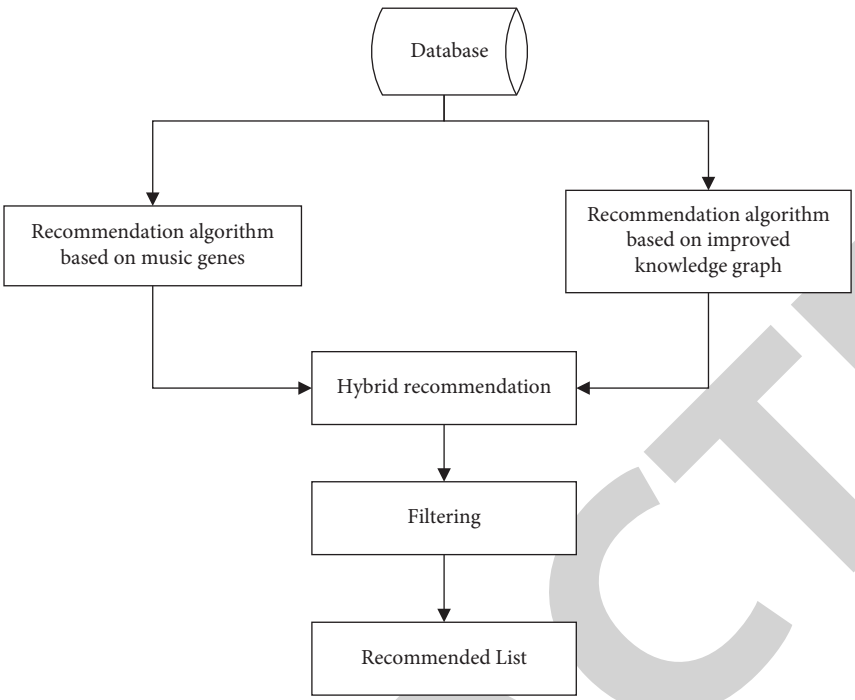

Figure 4: The overall flow chart of hybrid recommendation.

Table 1: Effect of 1different structures on hybrid recommendation algorithm.

Model	Dataset1		Dataset2	
	MAE	RMSE	MAE	RMSE
Model1	0.834	1.075	0.684	0.891
Model2	0.789	0.986	0.668	0.875
Model3	0.723	0.962	0.612	0.864

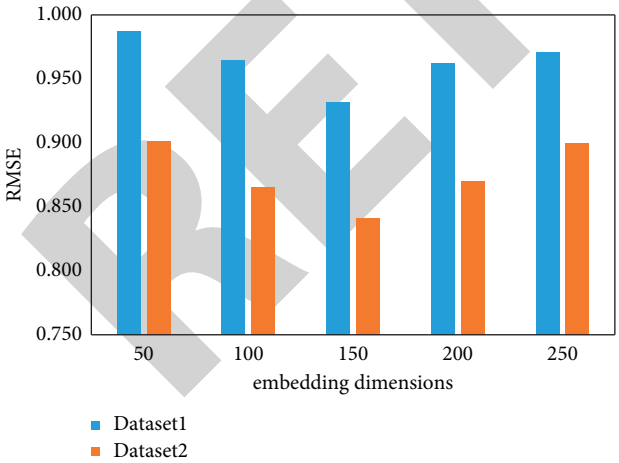

Figure 5: RMSE of the algorithm with different embedding dimensions.

From the experimental results, it can be seen that the root mean square error (RMSE) of the algorithm in this study first shows a decreasing trend as the embedding dimension of the knowledge graph representation learning increases. The RMSE value is the lowest when the embedding dimension is 150, and then increases as the embedding dimension increases. The increase of embedding dimension of the algorithm can better express the deep features between users and items, and the model can be better fitted and the recommendation effect is better. However, when the embedding dimension increases to a certain level, the features of users and items in the training set are too detailed, which makes the algorithm overfitting and the prediction effect becomes poor. Therefore, when the embedding dimension is 150, the recommendation effect is best.

In order to verify the effectiveness of the algorithm in this review, it is experimentally compared with five kinds of algorithms, including the label-aware recommendation algorithm and the recommendation algorithm combined with the knowledge graph. The descriptions of the compared algorithms are as follows:

(1) DSPR: Using label-based user and item information as the input of two neural networks, we capture the relevance of users and items using deep neural networks.

(2) TRSDL: Using pretrained word embedding to represent user labels, potential features of items and

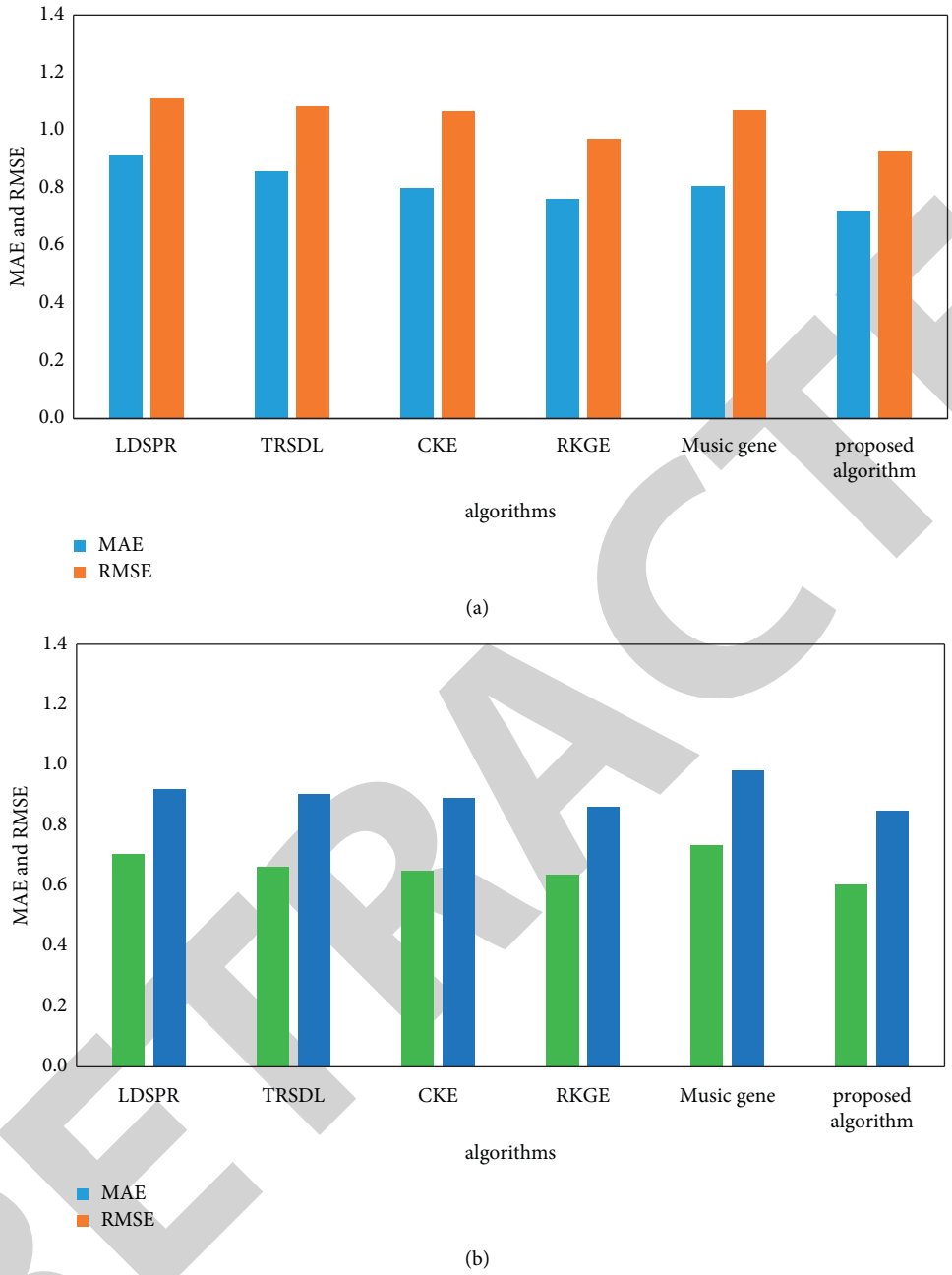

Figure 6: Experimental results of each algorithm on different datasets: (a) Experimental results on dataset1. (b) Experimental results on dataset2.

users are extracted for prediction using DNN and RNN, respectively.

(3) CKE: using a collaborative filtering based approach combined with knowledge graph embedding, items and user representations are learned through a single knowledge graph.

(4) RKGE: using heterogeneous information encoded by knowledge graphs, a recursive network structure is used to learn the relationship between users and items.

As can be seen from Figure 6, TRSDL has better prediction performance than DSPR in that the former takes into account the time factor in extracting label-based user latent features and captures the long- and short-term latent preferences of users through recurrent neural networks. CKE and RKGE also show better

performance, both of which use a single knowledge graph of the user's interaction history to construct a representation of the user, and then calculate the entity in the graph correlations with the items to be recommended, indicating that these models can use the knowledge graph to enhance user item information for effective recommendations. Compared with CKE, RGKE takes into account the semantic relationships between path-connected entities and uses recurrent networks to model the semantic paths of entities, thus providing a unified learning method for the representation of entities and entity relationships and improving the recommendation performance. On the other hand, it shows that the CKE method of training entity representations using methods such as TransR do not make good use of the information of the knowledge graph. Recommendation based on music gene does not take user project and user tag information into account, which leads to single recommendation result and small application scope, so the performance of music gene is not good. The hybrid recommendation algorithm proposed in this study outperforms other models on both datasets, which prove the effectiveness of the algorithm. The algorithm in this review combines label information with knowledge graph, uses item-oriented, and word-oriented knowledge graphs to enhance the knowledge of items and labels, respectively, and mines the deeper potential features of users and items through deep structure. At the same time, it can be seen that the two-layer LSTM designed in this review incorporating the attention mechanism plays an active role in the model and improves the prediction accuracy. Finally, the combination with music genes further improves the performance of music recommendation.

7. Conclusion

With the rapid evolution of music business, the music library is getting much richer, and the differentiation of users' preferences is getting much bigger as well. One of the difficulties in music business promotion nowadays is how to make accurate personalized recommendations to users from the huge music library conveniently and quickly. In this study, through analyzing the massive user behavior records kept in music websites, the hybrid recommendation algorithm based on music genes, and improved knowledge graphs utilizes the semantic information inherent in the items themselves and combines user tagging information, which can consider the attributes of items and users more comprehensively. Simultaneously, the algorithm uses two knowledge graphs to enhance the semantic information of items and user tags, and captures low-order and high-order features through a knowledge graph convolutional network. Finally, it combines the music gene recommendation model to make personalized music recommendations for users. The experimental results verify the effectiveness of the algorithm in this study. In future work, more efforts will be made to fuse multi-source heterogeneous auxiliary information and further improve the recommendation performance of the algorithm via extending embedding techniques and deep learning methods.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Shijiazhuang University.

References

[1] R. Kollolu, "Infrastructural constraints of Cloud computing," *International Journal of Management, Technology and Engineering*, vol. 10, pp. 255–260, 2020.

[2] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: A survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[3] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[4] X. Wen, "Using deep learning approach and IoT architecture to build the intelligent music recommendation system," *Soft Computing*, vol. 25, no. 4, pp. 3087–3096, 2021.

[5] Y. Xu and W. Zhou, "A deep music genres classification model based on CNN with Squeeze & Excitation Block," in *Proceedings of the 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 332–338, IEEE, Auckland, New Zealand, December 2020.

[6] M. S. Fathollahi and F. Razzazi, "Music similarity measurement and recommendation system using convolutional neural networks," *International Journal of Multimedia Information Retrieval*, vol. 10, no. 1, pp. 43–53, 2021.

[7] D. Xu, "Research on music culture personalized recommendation based on factor decomposition machine," *Personal and Ubiquitous Computing*, vol. 24, no. 2, pp. 247–257, 2020.

[8] D. Paul and S. Kundu, "A survey of music recommendation systems with a proposed music recommendation system," *Advances in Intelligent Systems and Computing*, Springer, Singapore, pp. 279–285, 2020.

[9] M. Pulis and J. Bajada, "Siamese neural networks for content-based cold-start music recommendation," in *Proceedings of the Fifteenth ACM Conference on Recommender Systems*, pp. 719–723, ACM, Amsterdam, The Netherlands, October 2021.

[10] D. Wu, "Music personalized recommendation system based on hybrid filtration," in *Proceedings of the 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pp. 430–433, IEEE, Changsha, China, January 2019.

[11] Y. Wang, "A hybrid recommendation for music based on reinforcement learning," in *Proceedings of the Pacific-Asia*

Conference on Knowledge Discovery and Data Mining, pp. 91–103, Springer, Singapore, May 2020.

[12] W. Feng, T. Li, H. Yu, and Z. Yang, "A hybrid music recommendation algorithm based on attention mechanism," in *Proceedings of the International Conference on Multimedia Modeling*, pp. 328–339, Springer, Prague, Czech Republic, May 2021.

[13] W. Wenzhen, "Personalized music recommendation algorithm based on hybrid collaborative filtering technology," in *Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pp. 280–283, IEEE, Xiangtan, China, August 2019.

[14] L. Wang, "Design and implementation of hybrid music recommendation system based on music gene," in *Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education*, pp. 121–124, ACM, Ottawa, Canada, October 2020.

[15] D. Hou, Z. Zhao, and S. Hu, "Multi-label learning with visual-semantic embedded knowledge graph for diagnosis of radiology imaging," *IEEE Access*, vol. 9, Article ID 15730, 2021.

[16] M. Alam, A. Gangemi, V. Presutti, and D. R. Recuperp, "Semantic role labeling for knowledge graph extraction from text," *Progress in Artificial Intelligence*, vol. 10, no. 3, pp. 309–320, 2021.