OPEN ACCESS
International Journal of Software Engineering
and Knowledge Engineering
Vol. 32, Nos. 11 & 12 (2022) 1753–1770

DOI: 10.1142/S0218194022500577

FAC: A Music Recommendation Model Based on Fusing Audio and Chord Features (115)

Weite Feng*, Junrui Liu†, Tong Li‡, Zhen Yang§ and Di Wu¶

Faculty of Information Technology
Beijing University of Technology
Beijing 100124, P. R. China
**fengwt@emails.bjut.edu.cn*
†*liujunrui@emails.bjut.edu.cn*
‡*litong@bjut.edu.cn*
§*yangzhen@bjut.edu.cn*
¶*wuxiaodou@emails.bjut.edu.cn*

Received 15 August 2022
Revised 7 September 2022
Accepted 9 September 2022
Published 27 October 2022

Music content has recently been identified as useful information to promote the performance of music recommendations. Existing studies usually feed low-level audio features, such as the Mel-frequency cepstral coefficients, into deep learning models for music recommendations. However, such features cannot well characterize music audios, which often contain multiple sound sources. In this paper, we propose to model and fuse chord, melody, and rhythm features to meaningfully characterize the music so as to improve the music recommendation. Specially, we use two user-based attention mechanisms to differentiate the importance of different parts of audio features and chord features. In addition, a Long Short-Term Memory layer is used to capture the sequence characteristics. Those features are fused by a multilayer perceptron and then used to make recommendations. We conducted experiments with a subset of the last.fm-1b dataset. The experimental results show that our proposal outperforms the best baseline by 3.52% on HR@10.

Keywords: Recommendation system; music information retrieval; chord; attention.

1. Introduction

With the rapid development of music streaming services, music recommendation has become an increasingly important topic, attracting the attention of both academia

‡Corresponding author.

and industry [1, 2]. Music content has been identified as useful information to promote the performance of music recommendations. Oord *et al.* use Convolutional Neural Network (CNN) to learn the mapping of audio content to the music vectors and then introduce these vectors into the music recommendation task [3, 4]. Some efforts have been made to hybridize the user's listening history with the audio content to generate recommendations. Lee *et al.* propose a user embedding approach that integrates user history records and CNN-based audio content with Neural Collaborative Filtering (NCF) [5] and generates recommendations end-to-end [6].

Given the success of audio features in tasks such as speech recognition, frequency domain features, such as Mel-frequency cepstral coefficients or spectrograms, are used to represent audio features in hybrid music recommendation algorithms [7]. However, although these low-level audio features are suitable for certain tasks, their discriminative power and semantics are limited [8, 9]. This makes them not fully suitable for music classification, musical emotion recognition, or music recommendation tasks, requiring a more meaningful representation of music [10]. Our previous study has disclosed that users tend to have different degrees of preference for different segments of music content, termed fine-grained music preferences [8]. However, existing embedding methods for music content do not distinguish between different parts of the music at a fine-grained level. Instead, they use CNN or Recurrent Neural Networks (RNN) to directly learn the mapping between the audio content and the embedding vector. Such coarse-grained embedding methods may trap existing methods into sub-optimal solutions.

We argue that users' fine-grained preferences on music content should be carefully mined through chord features. As a high-level music feature, a chord progression is a continuous sequence of chords (e.g. C-G-Am-F) that describes the structure of music, which is the defining feature on which melody and rhythm are built. To this end, we have proposed COAT in our previous work [9], which uses a user-based attention mechanism to learn users' fine-grained music preferences for different parts of the music and then has achieved better performance than the low-level audio feature. However, this work ignores the sequence characteristics in chords and does not use the audio features. There are great differences in the chord sequences of different styles of music. For example, the chord sequences of pop music usually contain fewer chords and have more repetitions between paragraphs, while the chord sequences of jazz usually contain more chords and little repetition between paragraphs.

In this paper, we extend our previous work [9] by fusing audio and chord features (FAC) to improve the music recommendation. Specifically, FAC uses a Generalized Matrix Factorization (GMF) layer to mimic matrix factorization and mine users' musical interests from their interaction with songs. In addition, FAC has two feature extractors to extract audio features and chord features, respectively. Audio features include some low-level time-domain-based or frequency-domain-based characteristics, which are helpful for music recommendations. To users' fine-grained music preferences for different parts of audio features, we use a user-based attention mechanism to deal with the fine-grained music preferences of users for music content

in the audio feature extractor. At the same time, the sequence characteristic inflects the trend of chord change in songs, so we use Long short-term memory (LSTM) to capture the sequence characteristics. Thus, there is not only a user-based attention mechanism but also LSTM in the chord feature extractor. Finally, a multi-layer perceptron (MLP) is used to fuse those feature vectors obtained from the two extractors and GMF and then make predictions. The main contributions of this paper are summarized as follows:

- We propose a recommendation model to fuse audio features and chord features. Based on these features, FAC can make better performance than only using chord features.
- We propose a chord feature extractor that models sequence characteristics and the relationship between user preference and chord features, both of which are essential for music recommendation.
- We propose an audio feature extractor that uses a user-based attention layer to learn the fine-grained music preferences of users for music content.
- We conducted experiments with a subset of the last.fm-1b dataset to assess the performance of our proposal. The experimental results show that our approach outperforms the baseline methods.

2. Related Work

In this section, we summarize the related music recommendation, attention, and pairwise-based methods, which are the basis of our method.

2.1. *Music recommendation algorithm*

Deep learning-based music recommendation approaches usually obtain a vector representation of a song from its audio content or metadata, known as an embedding vector. The obtained embedding vectors are then used to perform content-based recommendations, integrate into matrix factorization methods, or build hybrid music recommendation systems [7].

Oord *et al.* introduce deep learning techniques to music recommendation systems [3]. After obtaining the embedding vectors of users and songs by implementing a matrix factorization method, they train CNN to learn the mapping between the audio features and they embedding vector. This allows newly generated music to obtain its embedding vector via this CNN without interaction with the users. Beyond the audio content, scholars try to integrate information in more modalities. Yi *et al.* propose a cross-modal variable auto-encoder for content-based micro-video background music recommendations that integrates video content and audio content to form recommendations [11].

Since separating the process of acquiring music embedding vectors from acquiring user and song embedding vectors may produce sub-optimal solutions, some scholars

consider an end-to-end manner to build hybrid recommendation systems [6]. Liang *et al.* suggest a hybrid approach that first learns the content features through a multi-layer neural network and subsequently integrates them into the matrix factorization as a prior [12]. Lee et al. suggest a deep content-user embedding model, which learns users' and songs' embeddings through a multi-layer neural network while using CNN to learn audio features of songs, and combines the two in an end-to-end way to finally generate recommendations [6]. Feng *et al.* propose a hybrid music recommendation algorithm that combines user behavior and audio features to learn the fine-grained preferences of users for music content from multiple audio features by using an attention mechanism [8]. Valerio *et al.* use a hypergraph to model the relationship between users, songs, and tags, which is more likely to provide useful suggestions because that can understand the nucleus of the relationship between users and musics [2]. Yezi Zhang uses CNN to process spectrum and notes in musics [4]. The recommendation results are based on the similarity between customer preferences and the two musical features.

To sum up, much work has been done on integrating audio content into collaborative filtering recommender systems. However, these approaches have not yet explored the effects of higher-order music features with more explicit meanings in music recommendation tasks, nor have they been able to mine the fine-grained preferences of users for music content. The development of music information retrieval techniques and the application of attention mechanisms in recommender systems make it possible to fill this gap.

2.2. *Attention-based recommendation and data mining system*

The human attention mechanism inspires the attention mechanism in deep learning. Like the attention to a specific part of the input in human vision, applying the attention mechanism in recommender systems allows the model to filter the most informative part from the input features. Therefore reducing the influence of noisy data improves the effectiveness of the recommendation and brings some interpretability [13, 14].

Wang *et al.* propose a dynamic user modeling approach that introduces the attention mechanism into the collaborative filtering method [15]. The method accurately portrays user interests by combining temporal information from calculating the degree of influence of the K items that the user has recently interacted with. The incorporation of the attention mechanism enhances the effectiveness of the collaborative filtering method. Zhou *et al.* propose a framework based on self-attention for modeling user behavior. The introduced self-attention mechanism demonstrates better performance and efficiency in their experiments than CNN and RNN [16].

Du et al. introduce a user embedding-based attention mechanism in a sarcasm detection task, which allows the features of various aspects of the user to be used effectively [17]. For the next point-of-interest recommendation task, Liu *et al.* propose an attention-based category-aware GRU (ATCA-GRU) model [18].

The ATCA-GRU model can select the more significant parts of the relevant historical check-in trajectory to enhance the recommendation effect using the attention mechanism.

Gong *et al.* introduce an attention mechanism on the massive open online courses recommendation task [19]. They fuse meta-paths with contextual information by applying the attention mechanism on meta-paths of heterogeneous graphs to capture different students' different interests. Shi *et al.* propose a method based on meta-paths and the attention mechanism [20]. The attention mechanism differentiates the importance of different meta-paths, which improves the effectiveness of recommendations and brings some interpretability.

In summary, attention mechanisms have been widely used with good results in various data mining tasks, which allows us to apply them to the analysis of users' fine-grained music preferences.

2.3. *Pairwise-based recommendation methods*

General recommendation methods use a pointwise loss function, the MSE loss function, as their objective function. There is another loss function, the pairwise loss function, which is different from the pointwise loss function. Compared with the pointwise method that learns true labels, the pairwise method is designed to capture the existence of partial order among items in the original data. Bayesian personalized ranking (BPR) is a typical pairwise method and learns the global ranking of items from their local ranking relationships [21].

Some efforts have been made to apply pairwise loss functions to the content-based recommendation. For visual personalized ranking, VBPR [22] uses the pairwise loss function as its objective function to distinguish the important item embedding that is a high-level representation of visual features and extracted by CNN. The partial order information in pairwise loss functions can also be used to model the geographical neighbors by nearby centers, which can help point-of-interest recommendations [23]. Recently, the pairwise loss function has experimental improvement in training graph convolutional networks [24, 25].

3. Method

The architecture of our proposed FAC is shown in Fig. 1. FAC takes the one-hot vector of users and songs and the corresponding audio file of the song as input, and the output is the probability that the user listens to the song. FAC uses GMF to embed users and songs and learns the history of user-song interactions. Above this, we design two extractors to extract chord features and audio features, respectively. After obtaining the vectors representing the user's long-term interests and the vectors representing the music content, we use a stacked neural network (called a prediction layer) to learn the complex relationship between user behavior records and music content and thus generate recommendations.

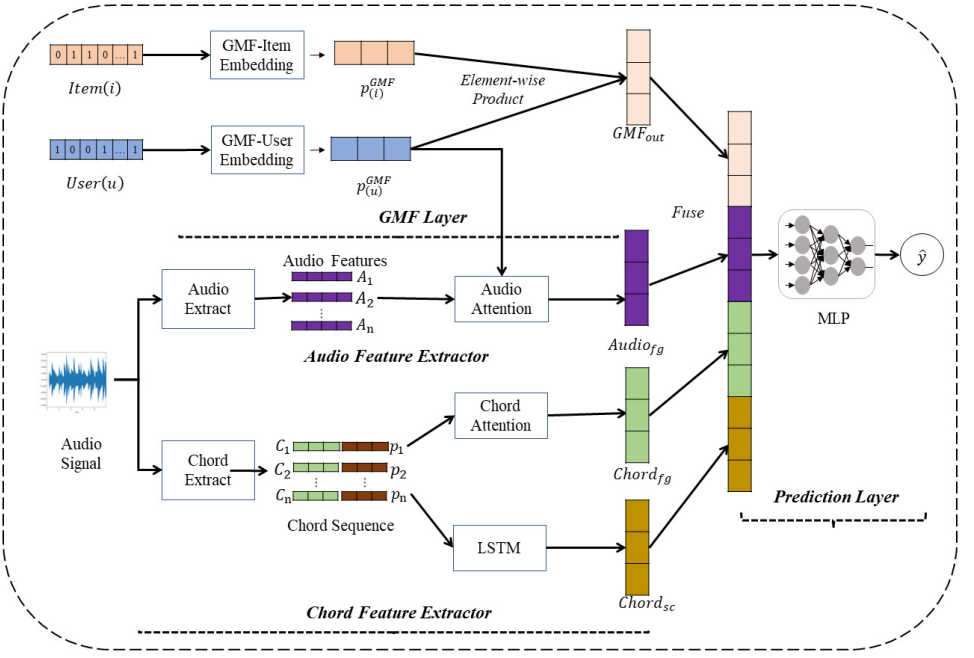


Fig. 1. The overall framework of our proposed FAC model.

3.1. *Generalized matrix factorization layer*

The user's interaction history is the most important representation of the user's interests, and mining the user's interest preferences can help improve the recommendation effect. Matrix factorization is a representative technique for this task, and being able to mimic this technique in a recommendation model is the foundation for building a successful recommendation model [5].

For this, we use a generalized matrix factorization layer (GMF) to mimic the matrix factorization. This layer first receives one-hot vector representations of the user and the song, denoted by V_u^U and V_i^I. After the embedding operation is implemented, these high-dimensional one-hot vectors are mapped to lower-dimensional vectors, called user and song embedding vectors. In the work of He *et al.* [5], after obtaining the embedding vectors for the user and the song, the probability of the user clicking on the song is obtained directly from the inner vector product. Here, we keep these two vectors and input them into the next part of the model. The function is as follows:

$$\mathrm{GMF}_{\mathrm{out}} = W_u^T V_u^U \odot W_i^T V_i^I, \tag{1}$$

where W_u and W_i are the learnable parameters that map V_u^U and V_i^I into embedding vectors, and $\odot$ denotes the element-wise product of vectors.

3.2. *Chord Feature Extractor*

There are three layers in chord feature extractor, i.e. chord embedding layer, LSTM layer and chord attention layer.

3.2.1. *Chord Embedding*

The chord feature extraction takes chord features from audio files. At this layer, we use a chord-extractor tool[a] to extract the chords of the music. As the number of chords is often inconsistent from song to song, and neural networks cannot handle variable-length data, we uniformly padded the collected chord sequences to 100 by repeating them in order.

After obtaining the chord sequence, we generate a random embedding vector representation for each chord. Each chord sequence can be represented by a matrix C, and C_i represents the ith chord vector in the chord matrix. The value of i indicates the order in which the chord appears in the time dimension.

3.2.2. *LSTM for sequence characteristics*

A chord, in music, is any harmonic set of pitches/frequencies consisting of multiple notes (also called "pitches") that are heard as if sounding simultaneously. Combinations of different chords also produce different effects. The sequence characteristic inflects the trend of chord change in songs and is very attractive to users. There are great differences in the chord sequences of different styles of music. Some special chord combinations always attract people's attention. Therefore, modeling sequence features can effectively improve the model performance. To capture the combined features between chords, we use LSTM which is a typical RNN and is usually used to model the sequence data such as sentences. The sequence characteristics learned by LSTM is

$$Chord_{sc} = LSTM(C). \tag{2}$$

3.2.3. *Chord attention layer*

LSTM can learn the sequence features of chords, but it lacks modeling of the user's personalized interest in chord features. Here we use a user-based attention layer to capture users' interests in chord features. And since different placements of the same chords produce different sounds (e.g. Am-F-C-G and C-G-Am-F are different chord progressions), we generate positional embedding for each position, representing as p_i.

Combining the user embedding vector e_u and the position vector p_i, we calculate the attention weight a_i of each chord embedding vector using the following equation:

$$a_i = \mathrm{h}^{\mathrm{T}} Relu\left(\mathrm{W}^{T}\begin{bmatrix}\mathrm{e_u}\\ \mathrm{C_i}\\ \mathrm{p_i}\end{bmatrix} + \mathrm{b}\right), \tag{3}$$

[a] https://ohollo.github.io/chord-extractor/.

where h, W and b are parameters, and *Relu* is the Relu activation function.

After applying Eq. (3) to calculate the individual attention weights a_i, we normalized them using softmax with the following equation:

$$\beta_{\mathrm{i}} = \frac{\exp(\mathrm{a_i})}{\sum_{\mathrm{j}=1}^{|\mathrm{C}|} \exp(\mathrm{a_j})}. \tag{4}$$

The normalized attention weights β_i represent the importance of the different parts of the song, with which we finally weighted and summed with the chord embedding vector to obtain the output of the chord attention layer $Chord_{fg}$. The user preference-based music content can be embedded as

$$Chord_{fg} = \sum_{i=1}^{|C|} \beta_i \cdot C_i. \tag{5}$$

3.3. *Audio feature extractor*

Existing embedding methods directly learn the mapping between the audio content and the embedding vector [3, 6]. Such coarse-grained embedding methods do not distinguish between different parts of the music at a fine-grained level that lead to a sub-optimal solution. Audio content directly reflects the content of music, so establishing a relationship between users and audio content will help achieve accurate music recommendations. Attention mechanism achieves excellent performance in some tasks [26]. Similar to the chord attention layer, the audio feature extraction uses a user-based attention layer to embed audio features. We denote the audio features of a song i as D_i, so that the representation of the embed result is

$$Audio_{fg} = \sum_{j}^{|D|} \beta_j * D_j, \tag{6}$$

where $\sum_j^{|D|}$ is a user-based attention score and is as follows:

$$\beta_j = \frac{W_u^T V_u^U \odot D_j}{\sum_j^{|D|} W_u^T V_u^U \odot D_j}. \tag{7}$$

3.4. *Prediction layer*

After obtaining user embedding and music content embedding from GMF and two feature extractors, we calculate the final listening probability using a stacked neural network with the following equation:

$$\hat{\mathrm{y}}_{\mathrm{ui}} = MLP\left(\begin{bmatrix} GMF_{\mathrm{out}} \\ Audio_{fg} \\ Chord_{fg} \\ Chord_{\mathrm{sc}} \end{bmatrix}\right), \tag{8}$$

where MLP stands for the common multi-layer perceptron, whose number of layers and shapes can be set flexibly. In this paper, we set its number of layers to 3 to avoid too many parameters causing overfitting. In terms of shape, we set it using a typical tower structure, where each layer has twice the number of neurons as the next layer. The premise of this approach is that setting smaller neurons in the higher-level neural network will enable more abstract information to be learned from the data [27].

3.5. *Objective function and optimal*

After feeding the model with information on chord sequences and audio features to represent the content of the music, the preference relationship between the user and the song becomes complex. Therefore, the model needs to have a stronger fitting capability to fit this kind of preference relationship. A pairwise loss function that has been identified that can effectively distinguish the importance of different features is used as FAC's objective [21, 25]. A pair of items $i >_u j$ indicates that a user u prefers item i to item j. For one user, given a positive sample item i and a negative sample item j, the predicted score of item i should be higher than that of item j [21]. The objective function is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{pair}} &= \max \sum_{u,i,j \in D} (\Theta | i >_u j) \\ &= \max \sum_{u,i,j \in D} \ln \sigma\big(\hat{p}_{ui} - \hat{p}_{uj}\big) - \lambda_\Theta \|\Theta\|^2, \end{aligned} \tag{9}$$

where σ is the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$, $\hat{p}_{ui}$ is the predicted score of user u on item i from a model, and λ_Θ is the hyperparameter of the regularization. The item j is sampled uniformly randomly.

4. Experiment Settings

In this section, we introduce the dataset used in the experiments. We then pose five research questions that we intend to answer in this paper to justify the proposed approach's effectiveness. Based on these questions, we design experiments and report their results.

4.1. *Dataset descriptions and constructions*

The experiments require a dataset containing the user's listening history and the song's audio file. For user listening history, we extracted a subset from the widely used last.fm-1b dataset [28]. As the dataset does not contain audio files of the songs, we downloaded the corresponding audio files from streaming platforms based on the collection of songs in the subset to form the dataset used in this paper.

Due to the size of the complete last.fm-1b dataset, conducting experiments on the complete data set would consume too much time. So we streamlined the last.fm-1b

dataset in the following steps: First, we used 2014 as the boundary to remove previous records from the complete dataset; Second, we filtered the top 10,000 popular songs from the song set to build a new subset; Finally, based on this subset, we removed listening records that were not relevant to it. We removed users with less than ten interaction records to ensure the dataset's quality. With 30,753 users, 10,000 songs and 1,533,245 interaction records, our dataset has a data sparsity of 99.50%.

4.2. *Research questions*

- RQ1: Whether the proposed method is better than traditional methods?
- RQ2: How does each extractor designed in FAC affect the results?
- RQ3: How many are the best negative sampling ratio for pairwise loss function?
- RQ4: Does attention is better than CNN to capture audio features?
- RQ5: How to fuse the chord features and audio features?

4.3. *Experiment design*

To address the above research questions, we design four experiments accordingly.

- **Experiment 1:** We use a comparative experiment to verify whether our proposed model can obtain better results than the baseline approach. The chosen baseline methods include a traditional matrix factorization algorithm, a neural network-based collaborative filtering algorithm, and hybrid music recommendation algorithms based on audio features or chord features.
- **Experiment 2:** To validate the effectiveness of two extractors, we conducted ablation experiments on FAC. We designed variants of the model with and without the designs and judged the effectiveness by comparing the performance of the recommendations.
- **Experiment 3:** FAC is trained by a pairwise loss function, which is influenced by its sampling ratio. Training data is construed by a user, a positive item, and a negative item. That a negative item is sampled from the items that the user does not interact with. We do an experiment to find the best ratio in the range from 1 to 10.
- **Experiment 4:** Previous works [3, 4, 6, 11] use CNN to deal with audio features. CNN has no interaction with users so it cannot inflect users' interest in audio features. This experiment gives evidence about the importance of modeling fine-grain features.
- **Experiment 5:** There are several fusion methods, like concat, mean, max. This experiment shows the best way to fuse the features that FAC learns.

4.3.1. *Parameter settings.*

The models involved in this paper use the same strategy for parameter settings. The range of searching for each hyperparameter is as follows: batch size is

[128,256,512,1024], learning rate is [0.0001, 0.0005, 0.001, 0.005] and embedding size is [8,16,32,64].

For the NeuMF method, the size of the predictive factor is the output dimension of the MLP and GMF layers in the method. For FAC, the size of the predictive factor is equal to the dimensions of the embedding vectors. For WMF, the number of predictive factors is equal to the embedding size. And for the LSTM-based approach and CNN-based approach, we use the Librosa [29] library to extract the MFCCs features from the audio to represent music content.

4.3.2. *Evaluation Protocols.*

We used a strategy called leave-one-out to test the model's effectiveness, which has also been widely adopted in other work [30]. Regarding this strategy, the test set consists of one positive sample and several negative samples, where the positive sample is the last song in the user's listening record. Given a user, it would be time-consuming that regard all non-interacted songs as negative samples and sort them. Thus, we only sample 99 songs that a user has not interacted with as negative samples. This is a common strategy [31]. We use two common evaluation metrics to measure the effectiveness of ranking, *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG) [32]. The HR metric is given a 1 or 0 depending on whether the positive sample appears in the final top-n list. NDCG gives finer scores to positive samples based on where they appear in the top-n list. Higher scores are given to positive samples that appear higher. We generate top-n recommendation lists for all users for each experiment round and use this to calculate two metrics, HR and NDCG. The average of all users' scores on both metrics is used as the final score of the model.

5. Results and Discussions

This section reports experimental results that demonstrate the effectiveness of the proposed method.

5.1. *Performance comparison (RQ1)*

In this experiment, we selected a traditional matrix factorization approach, a deep learning-based collaborative filtering approach, and a hybrid audio content approach as baselines for comparison with FAC model.

- **BPRMF** [21]: Bayesian personalized ranking matrix factorization is the first method that proposes the pairwise loss function for recommendation methods.
- **WMF** [33]: The method uses a weighted matrix factorization technique to obtain the embedding vector of users and items and produces recommendations from the inner vector product.

- **NeuMF** [5]: The method uses deep neural networks to implement collaborative filtering and is the basis for many neural network-based recommendation algorithms.
- **NeuMF with CNN**: On top of NeuMF, CNN is used to process MFCCs features as a way to compare the performance of our proposed chord attention layer with that of the CNN-based approach.
- **NeuMF with LSTM**: On top of NeuMF, LSTM is used to process MFCCs features as a way to compare the performance of our proposed chord attention layer with that of the RNN-based approach.
- **HRMA** [8]: HRMA is a hybrid method that uses an attention mechanism to model audio features, and uses a GMF layer and an MLP layer to model latent features.
- **COAT** [9]: COAT uses attention to model the fine-grained chord features and NeuMF to make recommendation.

Figures 2 and 3 show that the FAC model consistently achieves better results than the other methods on the two evaluation metrics. When embedding size is 64, FAC outperforms the best baseline COAT by 3.52% on HR@10. FAC improves performance because it models the sequence characteristics by an LSTM and audio features by a user-based attention mechanism. NeuMF performs slightly better than the COAT model when the predictive factor size is 8, because COAT is influenced by the noise from the music content when the embedding dimension is small. This finding suggests that mining users' behavioral history is crucial in designing recommendation algorithms. When the size of the neural network model is small, too much introduction may introduce more noise into the model and degrade the recommendation performance.

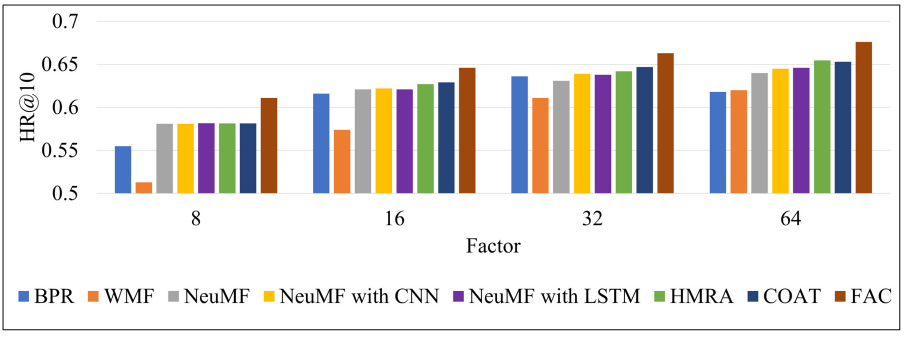


Fig. 2. HR@10 performance comparison of different methods.

Compared with NeuMF, the LSTM-based approach can obtain some improvement, while the CNN-based approach will obtain more inferior results. This phenomenon indicates that audio features in matrix form, though similar to pictures,

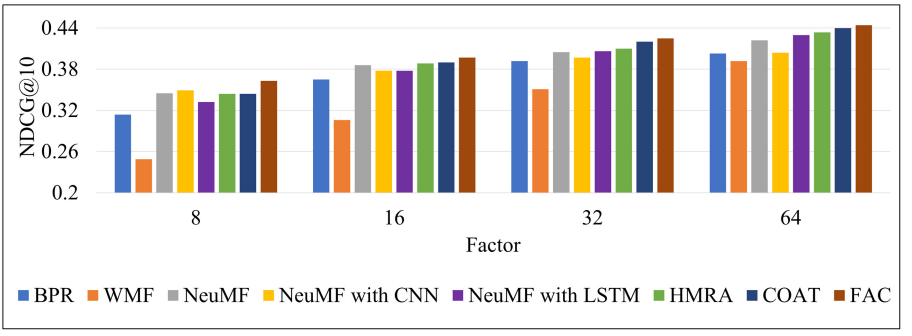


Fig. 3. NDCG@10 performance comparison of different methods.

have more significant temporal features than local features. And because low-level audio features are complex, it is challenging to process them properly in music recommendation models.

As shown in Figs. 2 and 3, the gap between the FAC model and other methods becomes larger as the predictive factor increases. In our analysis, the input chord sequence information makes the preference relationship between the user and the song more complicated, which means that the recommendation model needs to have stronger fitting power to learn this preference relationship. As the predictors increase, the model's width increases, making the model a stronger fitting capability. This phenomenon demonstrates again that when applying deep learning techniques to recommender systems, the use of larger-scale models brings improved recommendation results.

5.2. *Ablation experiment (RQ2)*

FAC uses LSTM and attention mechanism to get sequence characteristics, and audio features, respectively. We conduct ablation experiments to verify the effect of different modules on the final result. The $-Ca$ denotes that a model without the attention layer in a chord extractor, and $-A$ denotes that a model without an audio extractor.

Tables 1 and 2 show the results of the ablation experiments related to the attention mechanism proposed in this paper. As can be seen from the tables, FAC fuses

Table 1. Performance of various variants on HR@10.

Embedding size	8	16	32	64
COAT(-Ca)	0.392	0.491	0.556	0.608
COAT	0.581	0.629	0.647	0.653
FAC(-Ca-A)	0.502	0.594	0.641	0.662
FAC(-Ca)	0.514	0.590	0.638	0.665
FAC	**0.611**	**0.646**	**0.663**	**0.676**

Table 2. Performance of various variants on NDCG@10.

Embedding size	8	16	32	64
COAT(-Ca)	0.205	0.281	0.319	0.355
COAT	0.344	0.390	0.421	0.442
FAC(-Ca-A)	0.283	0.353	0.393	0.422
FAC(-Ca)	0.287	0.354	0.397	0.428
FAC	**0.363**	**0.397**	**0.425**	**0.444**

audio and chord features and achieves the best results. FAC(-Ca) is better than FAC (-Ca-A), indicating that audio features still have some potential features that FAC does not learn from chords. The results of either variant are not as good as FAC, indicating that music recommendation is a multi-feature fusion process. Incorporating more features can effectively improve the recommendation results.

5.3. *Best negative sampling ratio (RQ3)*

The negative sampling ratio is set from 1 to 10, influencing FAC by the pairwise loss function. The pairwise loss function maximizes the scores different between positive items and negative items. A big ratio will produce huge information about users' interests, i.e. users like some songs and dislike some songs. However, it could also make a mistake, i.e. it regards the positive one in test data as a negative one. Moreover, another potential shortcoming is that it conducts ineffective learning with a significant training time cost. So this experiment helps us to find the balance between time cost and model performance.

The experiment result is shown in Fig. 4. From it, we can see that the model performance first increases and then decreases as the sampling rate increases. The best HR@10 is achieved at the negative sampling rate is 6, and the best NDCG@10 is achieved at 4 and 6. At this point, the model can obtain the most useful information from the negative sampling method. As the sampling rate increase, the performance degrades. On the one hand, the model has a greater probability of treating the samples in the test set as negative samples, resulting in poor results. On the other hand, as the sampling rate increases, the model is more susceptible to the influence of

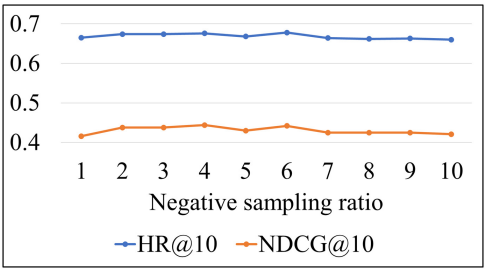
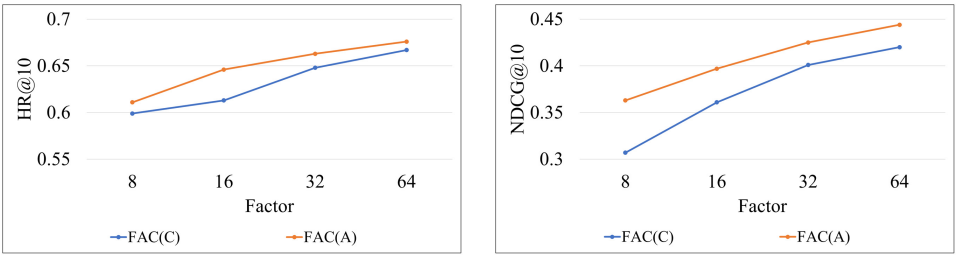


Fig. 4. The performance on different negative sampling ratio.

the data distribution [34, 35]. The model achieves the best performance when the negative sampling rate is 4.

5.4. *Attenion VS CNN (RQ4)*

To model user interest in audio features, FAC uses an attention mechanism. This experiment shows the importance of modeling fine-grain features from audio features. We compare the CNN-based FAC (FAC(C)) and attention-based FAC (FAC (A)) by different embedding sizes $8, 16, 32, 64$. The results are shown in Fig. 5.



(a) HR@10 performance on different embedding size

(b) NDCG@10 performance on different embedding size

Fig. 5. FAC performance comparison for processing audio features based on CNN and Attention.

FAC(A) outperforms FAC(C) under different embedding sizes. Music is essentially a complex audio signal. However, the creators are not too concerned with the performance of frequency characteristics. The creator triggers the creation of music from a global perspective. There is more of a global relationship between the audio features of music. The attention mechanism can learn the global dependencies between features, achieving better performance than CNN, which focuses only on local features. It can positively impact the effectiveness of recommendations that attention mechanisms are used to distinguish the importance of different parts from multimedia content in deep learning-based recommendation algorithms.

5.5. *Fusion Function Comparison (RQ5)*

To find the best way to fuse the features that FAC learns, we compare some fusion methods, including *concat*, *mean*, and *max*. The experiment result is shown in

Table 3. The performance with different fusion functions.

	HR@10	NDCG@10
Concat	**0.676**	**0.444**
Mean	0.674	0.433
Max	0.577	0.359

Table 3. The *concat* function achieves the best performance than other functions as it can distinguish the importance of different features by taking advantage of nonlinear functions. The *max* function emphasizes the most numerically significant features, not the most important ones, so its results are not good as others.

6. Conclusions and Future Work

In this paper, we propose fusing higher-order music features and lower-order frequency domain features to represent music content in the music recommendation model and differentiate the importance of music content properly. To this end, we propose a music recommendation model known as FAC. It uses two attention mechanisms to model chord features and audio features. In addition, LSTM is used to capture the sequence characteristics in chord features. The experimental results demonstrate the effectiveness of the method proposed in this paper. Also, the attention mechanism is better than CNN for capturing high-level features in audio.

FAC only uses audio features and chord features to make recommendations. There are still many higher-order music features, such as melody, rhythm, and lyrics, can be used to improve the model performance. At the same time, some deep learning models focus on enhancing the quality of interactions among multimodal embedding vectors, while FAC connects several feature embeddings simply. Thus, we will introduce more higher-order music features and design a feature interaction method to improve recommendation performance in the future.

Acknowledgments

This work is partially supported by the Project of Beijing Municipal Education Commission (No. KM202110005025), the Major Research Plan of National Natural Science Foundation of China (92167102), the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (CIT&TCD20190308), Beijing Natural Science Foundation Project (No. Z200002), and Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education. Weite Feng and Junrui Liu these authors contributed equally to this work.

References

1. M. Schedl, H. Zamani, C. Chen, Y. Deldjoo and M. Elahi, Current challenges and visions in music recommender systems research, *Int. J. Multimedia Inf. Retriev.* **7**(2) (2018) 95–116.
2. V. La Gatta, V. Moscato, M. Pennone, M. Postiglione and G. Sperl, Music recommendation via hypergraph embedding, *IEEE Trans. Neural Netw. Learn. Syst.* (2022) 1–13, Early Access.
3. A. van den Oord, S. Dieleman and B. Schrauwen, Deep content-based music recommendation, in *Advances in Neural Information Processing Systems*, eds. C. Burges,

L. Bottou, M. Welling, Z. Ghahramani and K. Weinberger, Vol. 26 (Curran Associates, Inc., 2013).
4. Y. Zhang, Music recommendation system and recommendation model based on convolutional neural network, *Mob. Inf. Syst.* **2022** (2022) 1–14.
5. X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T. Chua, Neural collaborative filtering, in *Proc. 26th Int. Conf. World Wide Web*, eds. R. Barrett, R. Cummings, E. Agichtein and E. Gabrilovich2017, pp. 173–182.
6. J. Lee, K. Lee, J. Park, J. Park and J. Nam, Deep content-user embedding model for music recommendation, arXiv:1807.06786.
7. M. Schedl, Deep learning in music recommendation systems, *Front. Appl. Math. Stat.* **5** (2019) 44.
8. W. Feng, T. Li, H. Yu and Z. Yang, A hybrid music recommendation algorithm based on attention mechanism, in *Proc. Multimedia Modeling - 27th Int. Conf. 2021, Part I*, eds. J. Lokoc, T. Skopal, K. Schoeffmann, V. Mezaris, X. Li, S. Vrochidis and I. Patras, Lecture Notes in Computer Science, Vol. 12572, 2021, pp. 328–339.
9. W. Feng, T. Li and Z. Yang, COAT: A music recommendation model based on chord progression and attention mechanisms, in *Proc. 34th Int. Conf. Software Engineering and Knowledge Engineering*, 2022, pp. 616–621.
10. P. Knees and M. Schedl, *Music Similarity and Retrieval - An Introduction to Audio- and Web-based Strategies*, The Information Retrieval Series, Vol. 36 (Springer, 2016).
11. J. Yi, Y. Zhu, J. Xie and Z. Chen, Cross-modal variational auto-encoder for content-based micro-video background music recommendation, arXiv:abs/2107.07268.
12. D. Liang, M. Zhan and D. P. Ellis, Content-aware collaborative music recommendation using pre-trained neural networks, in *Proc. 16th Int. Society for Music Information Retrieval Conf.*, 2015, pp. 295–301.
13. S. Zhang, L. Yao, A. Sun and Y. Tay, Deep learning based recommender system: A survey and new perspectives, *ACM Comput. Surv.* **52**(1) (2019) 5:1–5:38.
14. J. Chen, H. Zhang, X. He, L. Nie, W. Liu and T. Chua, Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention, in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, eds. N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries and R. W. White2017, pp. 335–344.
15. R. Wang, Z. Wu, J. Lou and Y. Jiang, Attention-based dynamic user modeling and deep collaborative filtering recommendation, *Expert Syst. Appl.* **188** (2022) 116036.
16. C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen and J. Gao, ATRank: An attention-based user behavior modeling framework for recommendation, in *Proc. Thirty-Second AAAI Conf. Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence, and the 8th AAAI Symp. Educational Advances in Artificial Intelligence*, eds. S. A. McIlraith and K. Q. Weinberger, 2018, pp. 4564–4571.
17. Y. Du, T. Li, M. S. Pathan, H. K. Teklehaimanot and Z. Yang, An effective sarcasm detection approach based on sentimental context and individual expression habits, *Cogn. Comput.* **14**(1) (2022) 78–90.
18. Y. Liu, A. Pei, F. Wang, Y. Yang, X. Zhang, H. Wang, H. Dai, L. Qi and R. Ma, An attention-based category-aware GRU model for the next POI recommendation, *Int. J. Intell. Syst.* **36**(7) (2021) 3174–3189.
19. J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang and P. S. Yu, Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view, in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, eds. J. X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen and Y. Liu2020, pp. 79–88.

20. B. Hu, C. Shi, W. X. Zhao and P. S. Yu, Leveraging meta-path based context for top- N recommendation with A neural co-attention model, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, eds. Y. Guo and F. Farooq2018, pp. 1531–1540.
21. S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in *UAI 2009, Proc. Twenty-Fifth Conf. Uncertainty in Artificial Intelligence*, eds. J. A. Bilmes and A. Y. Ng2009, pp. 452–461.
22. R. He and J. J. McAuley, VBPR: Visual Bayesian personalized ranking from implicit feedback, in *Proc. Thirtieth AAAI Conf. Artificial Intelligence*, eds. D. Schuurmans and M. P. Wellman2016, pp. 144–150.
23. S. Zhao, I. King and M. R. Lyu, Geo-pairwise ranking matrix factorization model for point-of-interest recommendation, in *24th Int. Conf., 2017, Proc. Part V Neural Information Processing*, eds. D. Liu, S. Xie, Y. Li, D. Zhao and E. M. El-Alfy, Lecture Notes in Computer Science, Vol. 10638, 2017, pp. 368–377.
24. H. Zhang and J. J. McAuley, Stacked mixed-order graph convolutional networks for collaborative filtering, in *Proc. 2020 SIAM Int. Conf. Data Mining*, eds. C. Demeniconi and N. V. Chawla2020, pp. 73–81.
25. Z. Liu, M. Wan, S. Guo, K. Achan and P. S. Yu, BasConv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network, in *Proc. 2020 SIAM Int. Conf. Data Mining*, eds. C. Demeniconi and N. V. Chawla, 2020, pp. 64–72.
26. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems*, eds. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Vol. 30 (Curran Associates, Inc., 2017).
27. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *2016 IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
28. M. Schedl, The LFM-1b dataset for music retrieval and recommendation, in *Proc. 2016 ACM on Int. Conf. Multimedia Retrieval*, eds. J. R. Kender, J. R. Smith, J. Luo, S. Boll and W. H. Hsu2016, pp. 103–110.
29. B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, librosa: Audio and music signal analysis in python, in *Proc. 14th Python in Science Conf.*, Vol. 8, 2015, pp. 18–25.
30. I. Bayer, X. He, B. Kanagal and S. Rendle, A generic coordinate descent framework for learning from implicit feedback, in *Proc. 26th Int. Conf. World Wide Web*, eds. R. Barrett, R. Cummings, E. Agichtein and E. Gabrilovich2017, pp. 1341–1350.
31. A. M. Elkahky, Y. Song and X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in *Proc. 24th Int. Conf. World Wide Web*, eds. A. Gangemi, S. Leonardi and A. Panconesi2015, pp. 278–288.
32. X. He, T. Chen, M. Kan and X. Chen, TriRank: Review-aware explainable recommendation by modeling aspects, in *Proc. 24th ACM Int. Conf. Information and Knowledge Management*, eds. J. Bailey, A. Moffat, C. C. Aggarwal, M. de Rijke, R. Kumar, V. Murdock, T. K. Sellis and J. X. Yu2015, pp. 1661–1670.
33. Y. Hu, Y. Koren and C. Volinsky, Collaborative filtering for implicit feedback datasets, in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 263–272.
34. S. Rendle and C. Freudenthaler, Improving pairwise learning for item recommendation from implicit feedback, *Seventh ACM Int. Conf. Web Search and Data Mining*, eds. B. Carterette, F. Diaz, C. Castillo and D. Metzler2014, pp. 273–282.
35. J. Liu, Z. Yang, T. Li, D. Wu and R. Wang, SPR: Similarity pairwise ranking for personalized recommendation, *Knowl. Based Syst.* **239** (2022) 107828.