Learning to embed music and metadata for context-aware music recommendation

Article *in* World Wide Web · December 2017

DOI: 10.1007/s11280-017-0521-6

CITATIONS
0

READS
17

4 authors, including:

Dongjing Wang
Zhejiang University
11 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Shuiguang Deng
Zhejiang University
121 PUBLICATIONS **1,148** CITATIONS
SEE PROFILE

Guandong Xu
University of Technology Sydney
197 PUBLICATIONS **881** CITATIONS
SEE PROFILE

World Wide Web
https://doi.org/10.1007/s11280-017-0521-6

Learning to embed music and metadata for context-aware music recommendation

Dongjing Wang[1,2] · Shuiguang Deng[1] · Xin Zhang[3] · Guandong Xu[2]

Received: 11 August 2016 / Revised: 7 May 2017 / Accepted: 13 December 2017

Abstract Contextual factors greatly influence users' musical preferences, so they are beneficial remarkably to music recommendation and retrieval tasks. However, it still needs to be studied how to obtain and utilize the contextual information. In this paper, we propose a context-aware music recommendation approach, which can recommend music pieces appropriate for users' contextual preferences for music. In analogy to matrix factorization methods for collaborative filtering, the proposed approach does not require music pieces to be represented by features ahead, but it can learn the representations from users' historical listening records. Specifically, the proposed approach first learns music pieces' embeddings (feature vectors in low-dimension continuous space) from music listening records and corresponding metadata. Then it infers and models users' global and contextual preferences for music from their listening records with the learned embeddings. Finally, it recommends appropriate music pieces according to the target user's preferences to satisfy her/his real-time requirements. Experimental evaluations on a real-world dataset show that the proposed approach outperforms baseline methods in terms of precision, recall, F1 score, and hitrate. Especially, our approach has better performance on sparse datasets.

✉ Shuiguang Deng
dengsg@zju.edu.cn

Dongjing Wang
tokyo1@zju.edu.cn

Xin Zhang
jiayou.zhangxin@163.com

Guandong Xu
Guandong.xu@uts.edu.au

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China

[2] Advanced Analytics Institute, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia

[3] School of Computer Science and Technology, Shandong University, Jinan, Shandong, China

Published online: 26 December 2017

Keywords Recommender systems · Music recommendation · Context-aware recommendation · Embedding

1 Introduction

Nowadays, there is an enormous amount of musical contents available on the Internet. For example, currently, Apple Music offers over 30 million songs.[1] Therefore, it becomes more and more difficult for people to find the music pieces that they really enjoy, which is known as the Paradox of Choice [25]. Therefore, recommender systems [19, 33, 38] have emerged to reduce the search costs and offer only the relevant items from enormous amounts of accessible data. Generally, traditional music recommender systems, such as collaborative filtering, content-based and hybrid approaches, try to solve the recommendation problem via the users' long-term music preferences.

However, people usually have different preferences and requirements under different contexts, and it has been proven that contextual information like physical surroundings, emotional state, time, presence of other people can help recommender systems better understand and satisfy the users' real-time requirements [14, 32]. Especially, music pieces are not neutral items but carriers of emotions and thoughts, and listening to music is a typical context-dependent behavior because people usually prefer different kinds of music under different contexts [16]. For instance, people generally prefer energetic music pieces with fast rhythm when doing exercises, and enjoy smoothing music pieces when resting. According to the classification in [2], there are three types of contexts in recommender systems: completely observable context, partially observable context, and unobservable context. In general, contexts in music recommendation, which our work focuses on, are partially observable or even unobservable. In addition, people can listen to music whenever and wherever they want, which makes the context of listening to music changeable and dynamic. Therefore, it becomes harder to acquire the real-time contexts of listening to music directly.

Fortunately, contexts of listening to music can be inferred from users' interactions with music systems. More specifically, contexts are reflected in the sequence of music pieces liked or listened to by the user in her/his current interaction with the system, such as recent playlists [13]. For example, in Pandora[2] (an online music streaming service website), users create different playlists by choosing different track seeds or artists. Then, users can play each of these playlists based on their current preferences which can be influenced by different contexts such as time, weather, emotion, or the task at hand. Given a set of music which the user plays or likes during an interaction, the recommender system should be able to recommend songs suitable for the current context of the user. Therefore, users' historical listening records indicate lots of information, such as the feature of music pieces, users' preferences for music, and a music recommender system should be able to infer the user's musical preferences from given music pieces liked or listened to by her/him and then recommend appropriate music pieces to satisfy her/his real-time requirements.

In this paper, we present a context-aware music recommendation approach, which can infer the user's global and contextual preferences from her/his listening records and recommend music pieces suitable for her/his current preferences for music. In detail, our approach consists of three steps. Firstly, the proposed approach learn the latent low-dimensional

[1] http://support.apple.com/en-us/HT204951

[2] http://www.pandora.com

representations (embeddings) of music pieces by considering music listening records and corresponding metadata. Note that, the learned embeddings can effectively capture music pieces' intrinsic features, and the music pieces that have similar features yield similar embeddings. Secondly, our approach infers and models the user's global and contextual preferences from her/his complete historical listening records and her/his active interaction session (music sequence recently played by a user) using the learned embeddings. Finally, music pieces, which conform to the user's global and contextual preferences, are recommended to satisfy her/his real-time requirements. Experimental evaluations show that the proposed approach has better performance than baseline approaches. Moreover, the results also show that our approach has a better ability to handle sparse data.

It is worthwhile to highlight the following contributions of the proposed recommendation approach in this paper.

- We propose a music embedding model to learn the real-valued, low-dimension embeddings of music pieces from music listening records and corresponding metadata.
- We propose a context-aware music recommendation method, which is able to obtain the users' global and contextual preferences for music and recommend appropriate music pieces to satisfy their real-time requirements.
- We conduct extensive experiments to evaluate the proposed method on real-world dataset collected from an online music service website. The results show that our method outperforms baseline methods, especially on sparse datasets.

The remainder of this paper is structured as follows. Section 2 describes the related works. In Sections 3 and 4, we introduce the motivation and the proposed approach in detail. Then, evaluations of the proposed approach are provided in Section 5. Finally, the conclusion and future work are given in Section 6.

2 Related work

In this section, we describe the related works on context-aware music recommendation, as well as works on embeddings which inspire our work.

2.1 Context-Aware music recommendation

Existing works on context-aware music recommendation can be divided into two categories according to the context types: environment-related context based approaches and user-related context based approaches.

2.1.1 Environment-related context based approaches

Such works are based on the fact that the environments have an influence on the users' state of mind or mood, and therefore influence the users' musical preferences [24]. For instance, people usually prefer different types of music in different seasons [27]. Consequently, music recommendation approaches with environment-related parameters perform better than those without considering contextual information. The environment-related contexts include time [10], location [17], weather [26] and hybrid context [40]. Kaminskas and Ricci [17] explored the possibilities of adapting music to the place of interests that the users are visiting. In [10], the authors incorporated temporal information in session-based collaborative filtering method to improve the performance of music recommendation. Park et al. [26] presented

a context-aware music recommender, which utilized several kinds of context information, including noise, light level, weather, and time. Hariri et al. [13] adopted an LDA model to infer the topic probability distribution of songs with tags and discovered a pattern of topics in the song sequences, which can be used as contexts to improve the performance of music recommendation.

2.1.2 User-related context based approaches

Compared with the environment-related contexts, the user-related contexts are the states of mind or moods of the users, and can therefore influence the users' musical preferences directly. The user-related contexts include activity [35], demographical information, emotional state [6, 9, 12], and hybrid context [41, 42]. Han et al. [12] proposed a context-aware music recommender system in which music is recommended according to the user's current emotion state and music's influence on changes of the users' emotion. Rho et al. [31] extracted the emotional information from music, including rhythm, scale and harmonics, and represented emotion as vector in the space of Thayer's emotion model. Then they used emotion vector as a supporting feature to compute music similarity. Deng et al. [9] presented another contextual music recommendation approach, which can infer users' emotion from her/his microblogs, and then recommend music pieces appropriate for users' emotion. Cai et al. [6] presented an approach named as MusicSense, which can infer users' emotion from Web documents read by the users and then match music to a document's content in terms of the emotions expressed by both the documents and the music. Yu et al. [41, 42] proposed context-aware recommendation models which consider hybrid context information (ranging from user preference and situation to device and network capability) as input for recommendation. Especially, their models can effectively perform multimedia content filtering, recommendation, and adaptation based on changing contexts.

2.2 Embedding

The proposed algorithm for learning the effective embedding of music pieces in this paper can be seen as part of the literature on representations learning [4]. In traditional representation learning models, each symbolic data, such as word and item, is represented as a feature vector using a one-hot representation. The object vectors have the same length as the whole object sets, and the position of the observed object in the vector representation is set as one. However, these models suffer from many problems, such as dimensional disaster and data sparsity, which limit their practicability to a great extent.

Neural models have been proposed to solve these problems mentioned above. These new models induce low dimensional embeddings of symbolic data by means of neural networks. Specifically, embedding is a kind of feature learning technique, where symbolic data are mapped from a space with one dimension per symbolic data object (one-hot representation) to a continuous vector space with much lower dimension based on training dataset, and the learned low dimensional representation of the object is called its embedding. Note that the learned embeddings can effectively capture items' important relationships and features in training dataset. Especially, in natural language processing (NLP) domain, neural models have been widely adopted to learn the effective embeddings of words and sentences [5]. Such models make use of the words ordered in sentences or documents, to explicitly model the assumption that the closer words in the word sequences are statistically more dependent. Although inefficient training of the neural network-based models has been an obstacle to their wider applicability in practical tasks when the vocabulary size grows to

several millions, this issue has been successfully addressed by recent advances in the field, particularly with the development of highly scalable skip-gram (SG) and continuous bag-of-words (CBOW) language models [22] for learning words' embeddings. These powerful, efficient models have shown very promising results in capturing both semantic and syntactic relationships between words in large-scale text corpora, and obtained state-of-the-art results on many NLP tasks. Recently, the concept of embedding has been expanded to many applications, including sentences and paragraphs representation [11], summarization [21], questions answering [43], recommender systems [34] and so on.

3 Motivation

Listening to music is a typical context-dependent behavior and the users usually prefer different types of music in different contexts. For example, a user may prefer sad music when experiencing bad mood and enjoy energetic music when working out. Therefore, contexts play an important role in predicting users' preferences for music and recommending appropriate music pieces. However, users can listen to music whenever and wherever they want, which makes it difficult to acquire the real-time contexts of listening to music directly. In fact, the contexts may not be captured with a static set of factors, but rather, it is dynamic and can be inferred from users' interactions with the system. More specifically, the contexts are reflected in the sequences of music pieces played or liked by the users in their current interactions with the system [13], such as recent playlists, so it is feasible to infer the contextual information from the users' listening behaviors. Furthermore, users' historical listening records indicate lots of information, such as the features of music pieces and the users' preferences for music, and a music recommender system should be able to infer the user's contexts and musical preferences from the given music pieces liked or listened to by her/him and recommend appropriate music pieces to satisfy her/his real-time requirements.

In detail, our work is based on the following three observations from the preliminary analysis of the users' listening data.

Observation 1: *every user has her/his own global musical preferences, which can be inferred from their music listening records* [7].

Every user has their own global musical preferences which are different from other users'. The global musical preferences are related to many factors, including the user' country, gender, age, personality, education, work, and so on. For example, teenagers may prefer listening to popular or rock music rather than classic music. Moreover, the users' global musical preferences can be inferred from their historical listening records, and then recommender systems can recommend appropriate music pieces.

Observation 2: *every user has different contextual musical preferences under different contexts* [18].

The users' general musical preferences may be diverse and various. However, people usually prefer only one or a few kinds of music pieces under certain contexts. For example, a user who likes both light music and hard rock music usually prefers the former when at rest. Therefore, context-aware recommender approach can generate better results by capturing and incorporating the users' contextual preferences than traditional approaches which do not consider contextual information. Although contextual preferences play an important role in music recommendation, it is usually dynamic and changeable, which makes it hard to acquire the real-time contexts directly.

Observation 3: *every user's contextual musical preferences usually maintain stable within a period (usually one kind of taste), which are reflected in their recent listening records* [13].

As mentioned above, every user's contextual preferences are usually certain and will maintain for a while. For example, a user may keep listening to sad music when experiencing bad mood, and this situation usually lasts for a period of time. Besides, most users are usually engaged in other things (also one kind of context) while listening to music, and they tend to listen to a list of music pieces with similar styles which conforms to their contexts. Therefore, it is feasible to infer the user's contextual preferences from music pieces in her/his active interaction session (music pieces recently listened to by her/him). On the other hand, users do not want to interrupt what they are engaged in to reselect music, which makes the precise prediction of users' contextual preferences more important.

Based on the three observations mentioned above, we need a model that is capable of (1) learning the embeddings of music pieces from music listening sequences, (2) inferring and modelling users' general and contextual musical preferences from her/his listening records, and (3) incorporating them into music recommendation.

4 Proposed approach

In this section, we introduce the task formalization of the proposed context-aware music recommendation approach, and then describe the proposed approach in detail, which consists of two components: music embedding model and context-aware music recommendation. Table 1 gives the basic symbols used in this paper.

Table 1 Basic symbols used in this paper

Symbol	Description
U	user set
u	a user
M	music set
m	a piece of music
H	all users' historical listening sequences
H^u	user u's historical listening sequence
m_i^u	the i-th music piece in user u's listening sequence H^u
$\mathbf{v}_{m_i^u}$	the embedding of music piece m_i^u
$\mathbf{p}_g^u$	user u's general music preference
$\mathbf{p}_c^u$	user u's contextual music preference
$p(m_i \mid u, \mathbf{p}_g^u, \mathbf{p}_c^u)$	the predicted preference of u to music piece m_i
$>_{u,\mathbf{p}_g^u,\mathbf{p}_c^u}$	the ranking of candidate music pieces

4.1 Formalization

Let $U = \{u_1, u_2, ..., u_{|U|}\}$ be a set of users and $M = \{m_1, m_2, ..., m_{|M|}\}$ be a set of music pieces, where $|U|$ and $|M|$ denote the total number of unique users and music pieces, respectively. For each user $u \in U$, her/his historical listening sequence is a list of music records (music pieces and playing timestamps), which is formally defined as $H^u = \{m^u_1, m^u_2, ..., m^u_{|H^u|}\}$, where $m^u_i \in M$ and $|H^u|$ is the length of u's listening sequence. Music pieces in each listening sequence are sorted according to the corresponding playing timestamps. Therefore, the task becomes to recommend music that user u would probably enjoy now given her/his historical listening sequence H^u.

There are two challenges here: (1) how to infer and model the users' global and contextual preferences for music from their historical listening sequences; (2) how to incorporate these preferences into music recommendation to satisfy the users' current requirements. To address these challenges, we first propose a music embedding model for learning the embeddings (feature vectors in low dimensional continuous space) of each music piece and each user. Then we propose a context-aware music recommendation approach, which can infer the user' global and contextual preferences for music, and take the preferences into consideration to generate appropriate recommendation.

4.2 Learning music embedding

Music Embedding Model (MEM) The Music Embedding Model (MEM) is proposed to learn the D-dimension real-valued embeddings (feature vectors) $\mathbf{v} \in R^D$ of each music piece m from all users' historical listening sequences $H = \{H^{u_1}, H^{u_2}, ..., H^{u_{|U|}}\}$, where $H^u = \{m^u_1, m^u_2, ..., m^u_{|H^u|}\}$ is user u's historical listening sequence.

The MEM is based on the three observations in Section 3. Firstly, each user usually has specific general musical preference (**Observation 1**), so the embeddings of the music pieces that are listened to by the same user should be similar to each other. Secondly, each user also has specific contextual musical preference usually maintains stable within a period (**Observation 2 and 3**), so the embeddings of the music pieces that are listened to by the same user within a period should be more similar to each other.

The graphical representation of MEM is shown in Figure 1. In this model, a sliding window is employed on the historical listening sequence of each user to generate the training data, where $2c$+1 is the length of the sliding window for listening sequences. Larger c results

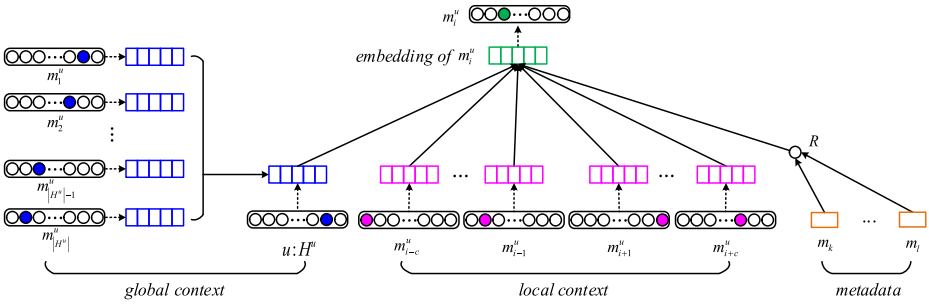

Figure 1 The framework of MEM

in more training examples, which leads to higher accuracy at the expense of more training time. Specifically, all music pieces before and after m_i in the sliding windows have strong relations to music piece m_i. In order to fully utilize these relations in the training dataset, MEM uses local context music pieces $\{m^u_{i-c} : m^u_{i+c}\}$ in each sliding window to predict the central music piece m^u_i. Except for the local context, we also incorporate the global context music pieces $u : H^u$ into our model. Therefore, the aim becomes predicting the central music piece m^u_i according to its local context music pieces $\{m^u_{i-c} : m^u_{i+c}\}$ and global context music pieces $u : H^u$. The corresponding probability function is defined as follows:

$$\Pr\left(m^u_i | m^u_{i-c} : m^u_{i+c}, u\right) = \exp\left(\bar{\mathbf{v}}^T \cdot \mathbf{v}'_{m^u_i}\right) \Bigg/ \sum_{m \in M} \exp(\bar{\mathbf{v}}^T \cdot \mathbf{v}'_m) \tag{1}$$

where $\mathbf{v}'_{m^u_i}$ is the output embedding of the central music piece m^u_i, and $\bar{\mathbf{v}}$ is the average input embedding of $u : H^u$ and $\{m^u_{i-c} : m^u_{i+c}\}$, which are the global and local context music pieces of m^u_i in u's historical listening sequence H^u, respectively. Formally, $\bar{\mathbf{v}}$ is defined as follows:

$$\bar{\mathbf{v}} = (\mathbf{v}_u + \sum_{-c \le j \le c, j \ne 0} \mathbf{v}_{m^u_{i+j}}) \Bigg/ (2c + 1) \tag{2}$$

where $\mathbf{v}_u$ is the average input embedding of all global context music pieces in u's historical listening sequence H^u. Specifically, $\mathbf{v}_u$ is defined as

$$\mathbf{v}_u = \sum_{1 \le i \le |H^u|} \mathbf{v}_{m^u_i} \Bigg/ \left|H^u\right| \tag{3}$$

where $|H^u|$ is the length of u's historical listening sequence H^u.

Then, the log-likelihood objective functions over the entire training data is defined as follows:

$$J = \sum_{u \in U, H^u \in H} \left(\sum_{m^u_i \in H^u} \log\Pr\left(m^u_i | m^u_{i-c} : m^u_{i+c}, u\right) \right) \tag{4}$$

Generally, music pieces are more similar with each other if they have similar metadata, including album and singer/player information. For example, if two music pieces are in the same album or sung/played by the same musician, they are very likely to have similar styles and genres. Therefore, the embeddings of music pieces that belong to the same album or sung/played by the same musician should be close to each other. Let $s(m_k, m_l)$ be the similarity score between music pieces m_k and m_l. Under the above assumption, we use the following heuristics to constrain the similarity score:

$$s\left(m_k, m_l\right) = \begin{cases} 1.0 & \text{if } a\left(m_k\right) = a\left(m_l\right) \text{ and } p\left(m_k\right) = p\left(m_l\right) \\ 0.5 & \text{if } a\left(m_k\right) = a\left(m_l\right) \text{ and } p\left(m_k\right) \ne p\left(m_l\right) \\ 0.5 & \text{if } a\left(m_k\right) \ne a\left(m_l\right) \text{ and } p\left(m_k\right) = p\left(m_l\right) \\ 0 & otherwise \end{cases} \tag{5}$$

where $a(m_k)$ and $p(m_k)$ denote the album and the musician of music piece m_k, respectively. If music piece m_k share the same musician and the same album with music piece m_l, their similarity score is 1.0. If music piece m_k only shares the same musician or the same album with music piece m_l, their similarity score becomes 0.5. Otherwise, their similarity score is 0. Note that the similarity parameters in $s(m_k, m_l)$ are set based on our experience, and if they are optimized on the dataset, the final performance can be further improved.

Then we encode the metadata, including album and musician information, using a regularization function R as follows:

$$R = \sum_{m_k \in M} \sum_{m_l \in M} s\left(m_k, m_l\right) \cdot \exp\left(\mathbf{v}_{m_k}^T \cdot \mathbf{v}_{m_l}\right) \tag{6}$$

where the similarity score $s(m_k, m_l)$ serves as a weighting function.

Therefore, we get the final objective function of MEM, which incorporates metadata information into the music embedding learning process, as follows:

$$J_{MEM} = J + \beta R \tag{7}$$

where β is the combination coefficient. Our goal is to maximize the combined objective function J_{MEM} over the entire training data.

Learning In the learning phase, we need to maximize the objective functions of the log probability defined in (7) over all users' historical listening sequences. However, the complexity of computing the corresponding soft-max function defined in (1) is proportional to the total music set size $|M|$. As $|M|$ can easily reach several millions, it is difficult to directly compute the probability. Two approaches of computationally efficient approximation of the full soft-max functions are hierarchical soft-max [23] and negative sampling [22]. In this paper, we adopt negative sampling to compute the objective function, which approximates the original soft-max function defined in (1) with the following formula:

$$\begin{aligned} \Pr\left(m_i^u | m_{i-c}^u : m_{i+c}^u, u\right) \; = \; & \log \sigma\left(\mathbf{v}_{m_i^u}^T \cdot \mathbf{v}'\right) \\ & + k \cdot \mathbb{E}_{m_{i'} \sim P_M}\left[\log \sigma\left(-\mathbf{v}_{m_{i'}}^T \cdot \mathbf{v}'\right)\right] \end{aligned} \tag{8}$$

where $\sigma(x) = 1/(1 + e^{-x})$, k is the number of negative samples, and $m_{i'}$ is the sampled music piece, drawn according to the noise distribution P_M, which is modeled by empirical unigram distribution over items. Negative sampling method generates k noise samples for prediction, in which k is a very small number compared with $|M|$. Therefore, the training time yields linear scale to the number of noise samples and becomes independent of the music set size $|M|$. Then stochastic gradient descent algorithm is used to maximize the optimized objective function represented by (8). Specifically, each embedding is firstly initialized to a D-dimension random vector, and the corresponding embeddings will be updated in the process of maximizing the objective functions with stochastic gradient descent algorithm. Finally, the embeddings of all users and music pieces are learned, and similar music pieces (or similar users) lie nearby in the D-dimension real-valued continuous space.

4.3 Context-aware music recommendation

Based on those learned embeddings, we can infer and model the users' global and contextual preferences from their music listening sequences.

Specifically, the user u's global preferences are reflected in u's music listening histories (**Observation 1** in Section 3), which can be obtained by averaging the embeddings of music pieces in her/his historical listening sequence H^u, which is formally defined as:

$$\mathbf{p}_g^u = \sum_{1 \le i \le |H^u|} \mathbf{v}_{m_i^u} \Big/ \left|H^u\right| \tag{9}$$

Besides, according to **Observation 2** and **Observation 3** in Section 3, u's contextual music preferences are reflected in u's recent music listening records, which can be obtained by averaging the embeddings of music pieces in her/his recent music listening sequence (music pieces in her/his active interaction session), which is defined as

$$\mathbf{p}_c^u = \sum_{m_j^u \in RS_u} \mathbf{v}_{m_j^u} \Big/ |RS_u| \tag{10}$$

where RS_u is u's recent listening sequence of music in her/his current interaction with the system, and $\mathbf{v}_{m_j^u}$ is the embedding of music piece m_j^u in RS_u.

Finally, a context-aware music recommendation approach is proposed to recommend appropriate music according to users' global and contextual music preferences. Formally, given a user u and her/his global and contextual music preferences $\mathbf{p}_g^u$ and $\mathbf{p}_c^u$, the predicted preference of u for music piece m_i is defined as

$$p\left(m_i|\mathbf{p}_g^u, \mathbf{p}_c^u\right) = \cos\left(\mathbf{v}_{m_i}, \mathbf{p}_g^u\right) + \cos\left(\mathbf{v}_{m_i}, \mathbf{p}_c^u\right) \tag{11}$$

where $\mathbf{v}_{m_i}$ is the learned embedding of music piece m_i and $\cos(\mathbf{v}, \mathbf{p})$ is the cosine similarity [3] of vectors $\mathbf{v}$ and $\mathbf{p}$.

Finally, the ranking of music pieces $>_{u,\mathbf{p}_g^u,\mathbf{p}_c^u}$ in our approach is defined as

$$m_i >_{u,\mathbf{p}_g^u,\mathbf{p}_c^u} m_i' :\Leftrightarrow p\left(m_i|\mathbf{p}_g^u, \mathbf{p}_c^u\right) > p\left(m_i'|\mathbf{p}_g^u, \mathbf{p}_c^u\right) \tag{12}$$

We then recommend the music pieces with high ranking scores to the target user.

5 Evaluation

In this section, we experimentally evaluate the performance of the proposed context-aware music recommendation method. In detail, we first describe the dataset, the baseline methods, and the experimental designs. Then we illustrate the embedding learned by music embedding model (MEM) with three examples. Next, we investigate how the dimension of the embeddings affect the performance of the proposed approach. This is followed by a subsection about the comparison between our method and baseline methods. Finally, we study how the proposed method and baseline methods perform on datasets with different sparsities.

All experiments are performed on an Intel Core i7-4700 base PC, which has 8GB RAM and runs a 64-bit Windows 10 operating system.

Table 2 Complete statistics of the dataset

#Users	#Music Pieces	#Listening	#Listening per user	#Listening per music
4,284	361,861	4,284,000	1,000	11.8

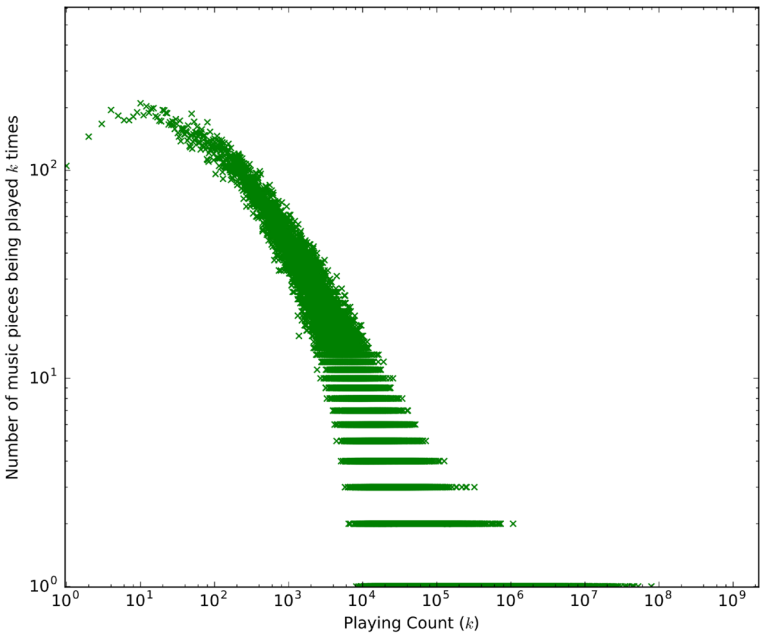

Figure 2 Popularity analysis of the dataset

5.1 Dataset

The dataset is collected from an online music service website named Xiami Music[3]. As shown in Table 2, the dataset[4] contains 4,284,000 music listening records of 4,284 users. Besides, every user has 1000 listening records and each music piece is interacted 11 times on average.

In addition, Figure 2 illustrates the relationship between popularity (listening count) and the number of music pieces with corresponding popularity. We can see that, only a small number of music pieces are very popular, while the majority of music are not so popular, which basically conforms to the power law distribution [1].

5.2 Baseline methods

In last two decades, many algorithms have been proposed for top-n recommendation on binary data without rating. Five state-of-the-art recommendation approaches, including Temporal Recommendation based on Injected Preferences Fusion (IPF) [39], Factorizing Personalized Markov Chains (FPMC) [29], Bayesian Personalized Ranking (BPR) [28],

[3]http://www.xiami.com

[4]Dataset link: https://zjueducn-my.sharepoint.com/personal/tokyo1_zju_edu_cn/_layouts/15/guestaccess.aspx?folderid=004419f09ac95493884d1f7314b89af43&authkey=AW1lUCePa24WA76yMxBJ4GM

FISMauc (FISM) [15] and user-based collaborative filtering method (UserKNN) [30] are used as baseline methods.

- **Temporal Recommendation based on Injected Preferences Fusion (IPF)**: IPF [39] is a novel context-aware approach that adopts Session-based Temporal Graph (STG) and personalized random walk algorithm to perform temporal-aware recommendation. Specifically, STG is a bi-partite graph which can efficiently capture and model the users' long-term and short-term preferences over time.
- **Factorizing Personalized Markov Chains (FPMC)**: FPMC [29] is a recommendation method based on personalized Markov chains over sequential (contextual) set data. Instead of using the same transition matrix for all users, this method uses an individual transition matrix for each user which in total results in a transition cube.
- **Bayesian Personalized Ranking (BPR)**: BPR [28] is a recommendation method based on a generic optimization criterion BPR-Opt for personalized ranking that is the maximum posterior estimator derived from a Bayesian analysis of the recommendation problem, and a corresponding generic learning algorithm named LearnBPR that is based on stochastic gradient descent with bootstrap sampling.
- **FISMauc (FISM)**: FISM [15] is an item-based recommendation method for generating top-n recommendations that learns the item-item similarity matrix as the product of two low dimensional latent factor matrices. Specifically, these matrices are learned using a structural equation modeling approach, wherein the value being estimated is not used for its own estimation.
- **User-based collaborative filtering method (UserKNN)**: UserKNN [30] is a classical collaborative filtering recommendation method.

5.3 Experimental designs

In this section, we introduce the detailed experimental designs, including dataset partition, evaluation metrics as well as settings of parameters.

5.3.1 Dataset partition

As mentioned in Section 4.1, the historical listening sequence of each user $u \in U$ in the collected dataset is a list of music records sorted according to their playing timestamps, which is formally defined as $H^u = \{m_1^u, m_2^u, ..., m_{|H^u|}^u\}$, where $m_i^u \in M$ and $i \in [1, |H^u|]$. In addition, each user's historical listening sequence H^u can be aggregated into sessions $S^u = \{S_1^u, S_2^u, ..., S_{|S^u|}^u\}$, where music pieces with close playing timestamps are grouped into the same session. Formally, u's n-th session is defined as $S_n^u = \{m_{n,1}^u, m_{n,2}^u, ..., m_{n,|S_n^u|}^u\}$, where $m_{n,i}^u \in M$ and $i \in \left[1, \left|S_n^u\right|\right]$. For example, as shown in Table 3, u's listening sequence contains 9 pieces of music and the corresponding timestamps. Obviously, the first four pieces can be aggregated into the same session because their playing timestamps are close to each other. Similarly, the other five pieces of music are aggregated into another session. More formally, the session set of u is $S^u = \left\{S_1^u, S_2^u\right\}$, where $S_1^u = \left\{m_1^u, m_2^u, m_3^u, m_4^u\right\}$ and $S_2^u = \left\{m_5^u, m_6^u, m_7^u, m_8^u, m_9^u\right\}$.

The goal of this experiment is to evaluate the performance of different recommendation approaches in making good recommendation given the users' historical listening sequences. Therefore, we can split the whole dataset into training sets and test sets according to the idea of 10-fold cross-validation. In each validation, we keep the complete listening records of 90% users and the first half of each session in the remaining 10% users' historical listening

Table 3 Listening sequence of u

No.	Music name - Artist	Playing time	H^u	S^u
1	Hero - Mariah Carey	2015-09-23 19:55	m_1^u	S_1^u
2	My Love (Live) - Celine Dion	2015-09-23 19:59	m_2^u	S_1^u
3	My Heart Will Go On - Celine Dion	2015-09-23 20:07	m_3^u	S_1^u
4	Living For Love C Madonna	2015-09-23 20:12	m_4^u	S_1^u
5	Stairway to Heaven - Led Zeppelin	2015-09-24 10:35	m_5^u	S_2^u
6	Knockin' on Heaven's Door - Guns N' Roses	2015-09-24 10:43	m_6^u	S_2^u
7	Enter Sandman - Metallica	2015-09-24 10:49	m_7^u	S_2^u
8	Nothing Else Matters - Metallica	2015-09-24 10:54	m_8^u	S_2^u
9	Master of Puppets - Metallica	2015-09-24 11:01	m_9^u	S_2^u

sequences as the training set, and use the following half of each session (test session) for the remaining 10% users as the test set.

5.3.2 Evaluation metrics

The performance is evaluated for each test session T in the test set. For each recommendation, we generate a list of n music pieces, denoted by R. The following four metrics [3, 8] are used to evaluate the performance of all recommendation approaches.

1. **Precision**

   Precision (also called positive predictive value) is the fraction of recommended music pieces that the target user actually listened to. Its definition is given below:

$$Precision = \sum_{1 \le i \le \#(recs)} |R_i \cap T_i| \,/\, |R_i|$$

   where:

   - $\#(recs)$ is the total number of recommendations.
   - R_i is the recommended music list of the i-th recommendation.
   - T_i is the music list of the i-th recommendation in the test data, which is actually listened to by the users.

2. **Recall**

   Recall (also known as sensitivity) is the fraction of interested music of the target user that are recommended, and its definition is given as:

$$Recall = \sum_{1 \le i \le \#(recs)} |R_i \cap T_i| \,/\, |T_i|$$

   where:

   - $\#(recs)$ is the total number of recommendation.
   - R_i is the recommended music list of the i-th recommendation.
   - T_i is the music list of the i-th recommendation in the test data, which is actually listened to by the users.

Table 4 Parameter settings for training MEM

Parameter	Value	Description
β	0.01	The weight of metadata in MEM
Dimension	[50, 300]	The dimension of the learned embeddings
Window $(2c+1)$	5	The number of the music pieces in the context
Negative sample	20	The number of "noise items" should be drawn (in order to increase the efficiency of the training progress)
Down sample	1e-5	Higher frequency items are randomly down sampled
Min-count	1	Items that appear less than the min-count value are ignored
Iteration	10	The count of training iteration

3. **F1 Score**

   F1 score is the harmonic mean of precision and recall. It is formally defined as follows:

$$F1\ score = 2 \times Precision \times Recall \big/ (Precision + Recall)$$

4. **Hitrate**

   Hitrate is the fraction of hits, and a hit means the recommendation list contains at least one music pieces that the user actually listened to. For example, as for a line (u, m) in the test data, if the recommended list of u contains m, then it is a hit. The definition is given below:

$$HitRate = \#(hits) \big/ \#(recs)$$

   where:

   – $\#(hits)$ is the total number of hits.
   – $\#(recs)$ is the total number of recommendation.

5.3.3 Parameter settings

The detailed configurations of the parameters in MEM and the corresponding descriptions are given in Table 4 as follows.

Specifically, the combination weight β used in MEM plays an important role in producing high quality music embedding. Overemphasizing the weight of the original objective may result in weakening influence of metadata, while putting too large weight on metadata may hurt the generality of the learned music embeddings. According to the experiments in Table 5, we set $\beta = 0.01$. Besides, the window size $(2c + 1)$ also play an important role

Table 5 Parameter β's impact on F1 Score and Hitrate

β	F1 Score@5	Hitrate@5
0	5.21%	27.83%
0.001	8.76%	47.39%
0.005	9.54%	48.94%
0.01	**9.99%**	**50.60%**
0.05	8.72%	44.69%
0.1	7.13%	36.15%

Table 6 Parameter c's impact on F1 Score and Hitrate

c	F1 Score@5	Hitrate@5
1	7.54%	41.94%
2	9.99%	50.60%
3	10.12%	51.09%
4	10.24%	51.36%
5	10.31%	51.48%

in producing high quality music embedding. As shown in Table 6, larger c results in more training examples, which leads to higher accuracy at the expense of more training time. Finally, we set the window size $c = 2(2c + 1 = 5)$.

Moreover, the dimension varies from 50 to 300, and we will explore the optimal value by preliminary experiments in Section 5.5.

5.4 Illustration of MEM's effect

In order to show what the learned embeddings look like, some illustrations of the learned embeddings are given before the evaluations of our approach.

5.4.1 Illustrations of artists' embeddings

We firstly analyze the embeddings of some selected artists' music pieces with t-SNE [20], which can visualize high dimensional data. More specifically, Table 7 shows several well-known artists and their tag information which are collected from last.fm[5], and Figure 3 shows the 2-dimensional single-point embeddings of top 10 music pieces of each artist with t-SNE. From the results, we can draw several conclusions.

Firstly, it is interesting to observe that music pieces by the same artist cluster tightly. The reason is two-fold. On the one hand, each singer's specific styles are reflected in the users' music listening sequences as well as the metadata of music pieces, which conforms to the three observations mentioned in Section 3. On the other hand, our music embedding model can effectively learn the accurate embeddings of music pieces from music listening sequences and the metadata of music pieces.

Secondly, the music pieces that are sung/played by the artists of similar genres lie nearby in the 2-dimension space. For example, Gun N' Rose, Bon Jovi, and Bob Dylan (1, 3, and 4) are three famous rock singers from Europe or the United States, and the embeddings of their music pieces are close to each other in the 2-dimension space. Besides, both Maroon 5 and Robbie Williams (2 and 5) have styles of alternative rock and pop, so the embeddings of their music pieces lie between the embedding of classic pop music and the embedding of rock music. Moreover, Lady Gaga, Adele, and Mariah Carey (7, 8, and 9) are three famous female vocalists of pop styles, and the embeddings of their music pieces are also close to each other in the 2-dimension space.

Thirdly, some slight differences in styles are also reflected in the learned embeddings, which further demonstrates the effectiveness of the MEM. For example, both Joe Hisaishi and Yuki Kajiura (10 and 11) are Japanese instrumental soundtrack masters, and their pieces' embeddings are closer to each other than the embeddings of other artists' pieces

[5] http://www.last.fm

Table 7 Basic information of some famous artists

No.	Artist	Tags in last.fm
1	Guns N' Roses	rock, hard rock, classic rock, metal,80s
2	Maroon 5	pop, rock, pop rock, alternative, alternative rock
3	Bon Jovi	rock, hard rock, classic rock, hair metal, 80s
4	Bob Dylan	folk, rock, folk rock, classic rock, songwriter, 60s
5	Robbie Williams	pop, British, britpop, rock, alternative rock
6	Justin Bieber	pop, black metal, rnb, hip-hop, r&b
7	Lady Gaga	pop, dance, electronic, epic, female vocalists
8	Adele	pop, soul, British, songwriter, female vocalists
9	Mariah Carey	pop, rnb, soul, female vocalists, 90s
10	Joe Hisaishi	sound track, Japanese, instrumental, anime, classical, piano
11	Yuki Kajiura	sound track, Japanese, instrumental, anime, j-pop

in the 2-dimension space. However, Hisaishi's soundtracks are piano pieces with classical styles while Yuki Kajiura's soundtracks are j-pop styles, so there exists a little distance between the embeddings of their pieces in the 2-dimension space.

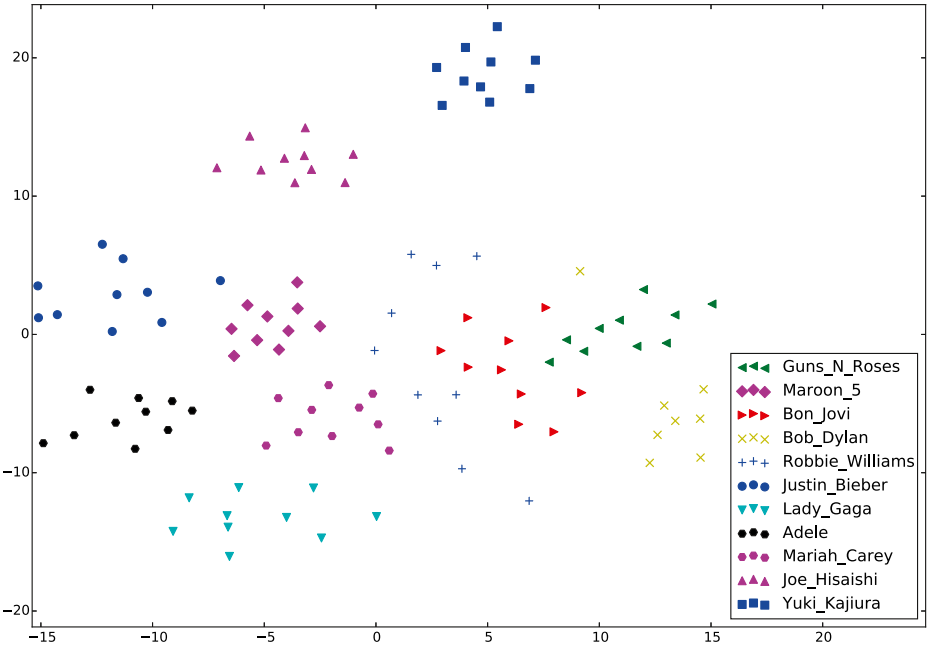

Figure 3 Visual representation of embedding of songs from selected artists in 2-dimension

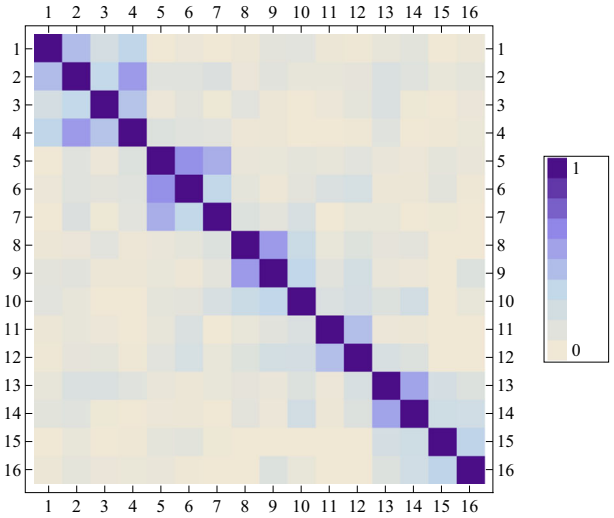

Figure 4 Similarity visualization of music examples with the embedding

5.4.2 Illustrations of selected music pieces' embeddings

While the visualization in Figure 3 provides interesting qualitative insights about artists, we now provide a further quantitative display of some selected music pieces with different styles. Figure 4 shows the visualization of similarity among music examples given in

Table 8 Basic information of selected music pieces

No.	Song - Artist	Tags
1	Drowning - Backstreet Boys	pop, ballad, boy bands
2	As Long as You Love Me - Backstreet Boys	pop, boybands, 90s
3	Swear It Again C Westlife	pop, Irish, 90s
4	My Love C Westlife	pop, boy bands, Irish
5	Don't Cry - Guns N' Roses	classic rock, hard rock, ballad
6	Knockin' on Heaven's Door - Guns N' Roses	rock, classic rock, hard rock
7	Fade to Black - Metallica	rock, thrash metal, heavy metal
8	Fall Again - Kenny G	smooth jazz, R&B, Soul
9	Heart and Soul - Kenny G	smooth jazz, Rhythm and blues
10	I Believe - Dave Koz	jazz, smooth jazz, saxophone
11	The Look of Love - Diana Krall	jazz, smooth jazz, vocal jazz, female vocalists
12	Don't Know Why - Norah Jones	jazz, blues, female vocalists
13	Summer - Joe Hisaishi	sound track, Japanese, anime, instrumental, classical
14	Moonlit Sea of Clouds - Joe Hisaishi	Sound track, anime, classical, instrumental, Japanese
15	Canta Per Me - Yuki Kajiura	sound track, anime, Japanese
16	zero hour - Yuki Kajiura	sound track, anime, Japanese

Table 9 5-dimension embeddings of selected music pieces

No.	Song - Artist	Embedding
1	Drowning - Backstreet Boys	(−0.291412, 0.270251, −0.639598, 0.32234, −0.13059)
2	As Long as You Love Me - Backstreet Boys	(−0.478884, 0.592784, −0.63983, 0.200698, −0.139833)
3	Swear It Again C Westlife	(−0.218441, 0.429761, −0.578103, 0.149284, −0.016236)
4	My Love C Westlife	(−0.568655, 0.775186, −0.400681, 0.001539, −0.34142)
5	Don't Cry - Guns N' Roses	(0.033134, 0.41475, −0.667938, −4.06E-4, −0.386768)
6	Knockin' on Heaven's Door - Guns N' Roses	(−0.130283, 0.361255, −0.62779, −0.03417, −0.426643)
7	Fade to Black - Metallica	(0.366062, 0.245874, −0.60944, 0.015976, −0.373623)
8	Fall Again - Kenny G	(−0.100747, 0.780737, −1.538182, 0.71393, −0.50115)
9	Heart and Soul - Kenny G	(−0.011782, 0.836092, −1.271244, 0.665455, −0.249353)
10	I Believe - Dave Koz	(0.398667, 0.477762, −0.662803, 0.714214, −0.25805)
11	The Look of Love - Diana Krall	(−0.179568, 0.095475, −1.014773, 0.036958, −0.375554)
12	Don't Know Why - Norah Jones	(−0.032461, 0.516513, −0.719967, 0.110694, −0.531966)
13	Summer - Joe Hisaishi	(0.242241, 0.724721, −0.471825, 0.516915, 0.133137)
14	Moonlit Sea of Clouds - Joe Hisaishi	(0.180384, 0.571349, −0.13619, 0.702893, 0.577625)
15	Canta Per Me - Yuki Kajiura	(−0.151482, 0.138815, −0.217584, 0.820246, 0.201598)
16	zero hour - Yuki Kajiura	(−0.024295, 0.654515, 0.127545, 1.551609, 0.222309)

Table 8. Besides, Table 9 gives the 5-dimension real-valued embeddings of the selected music pieces in Table 8. From the results, we can draw three conclusions.

Firstly, music pieces with similar styles and genres have similar embeddings. For example, the embeddings of the last four anime soundtrack music pieces (13-16) composed by Japanese artists in Table 8 are indeed similar to each other than the other music pieces.

Secondly, the embeddings of music pieces sung/played by the same artists are usually closer to each other than the embeddings of other pieces. For example, none of these four music pieces (13-16) are similar with the other music pieces (1-12) in Table 8.

Thirdly, some slight differences in styles and genres of music pieces are also shown by the learned embeddings, which shows that the learned embeddings by MEM can effectively capture the accurate features of the corresponding music pieces. For example, as for the last four music pieces (13-16), all of which are soundtracks for anime, and they are more similar to each other than the other pieces (1-12) in Table 8. In addition, the former two pieces (13-14) are more similar to each other than the latter two pieces (15-16) in Table 8. The reason is that Hisaishi's soundtracks are instrumental pieces with classical styles while Yuki Kajiura's soundtracks are not.

5.4.3 Illustrations of the embeddings of users' listening records

While the visualization in Figure 4 provides interesting qualitative insights about artists, we now provide a further quantitative display of some selected users. Figure 5 gives the visualization of the embeddings of different users' listening records. From the results, we can draw two conclusions. Firstly, the music pieces listened to by each user form one or several clusters, which shows that users have different general preferences for music, and they usually enjoy one or several specific kinds of music (**Observation 1**). For example, user1 has relatively focused preferences while user3 has a broader range of interests. Secondly,

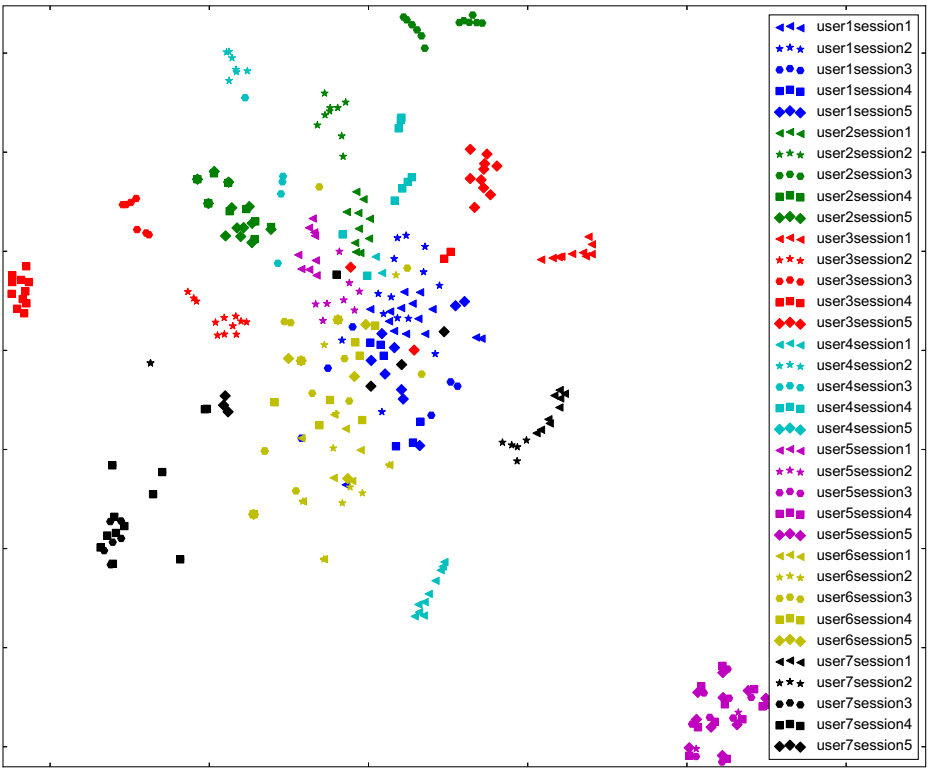

Figure 5 Visualization of the embeddings of different users' listening records

the music pieces in each session cluster tightly, which shows that each user has different contextual preferences for music under different contexts (**Observation 2-3**).

In conclusion, the illustrations confirm our observations mentioned in Section 3 and show that the recommending strategy of incorporating both user's global and contextual preferences is reasonable and sound. On the other hand, the illustration also shows that the embeddings learned by our method from music listening sequences depict the intrinsic features of music pieces effectively and are useful for many other tasks, such as similarity measure, corpus visualization, automatic tagging, and classification.

5.5 The impact of dimension

The dimension of the embeddings is very important in music recommendation, and it is necessary to choose a proper dimension to balance the performance of accuracy and efficiency. Specifically, the embeddings of higher dimension can capture more useful features and depict music pieces better. On the other hand, the learning process needs more computation resources, and our recommendation task does not need embeddings of too high dimension. In order to investigate how the embedding's dimension affect the performance of the proposed approach, we evaluate our method with different dimensions (50, 100, 150, 200, 250 and 300), and the results are shown in Figure 6.

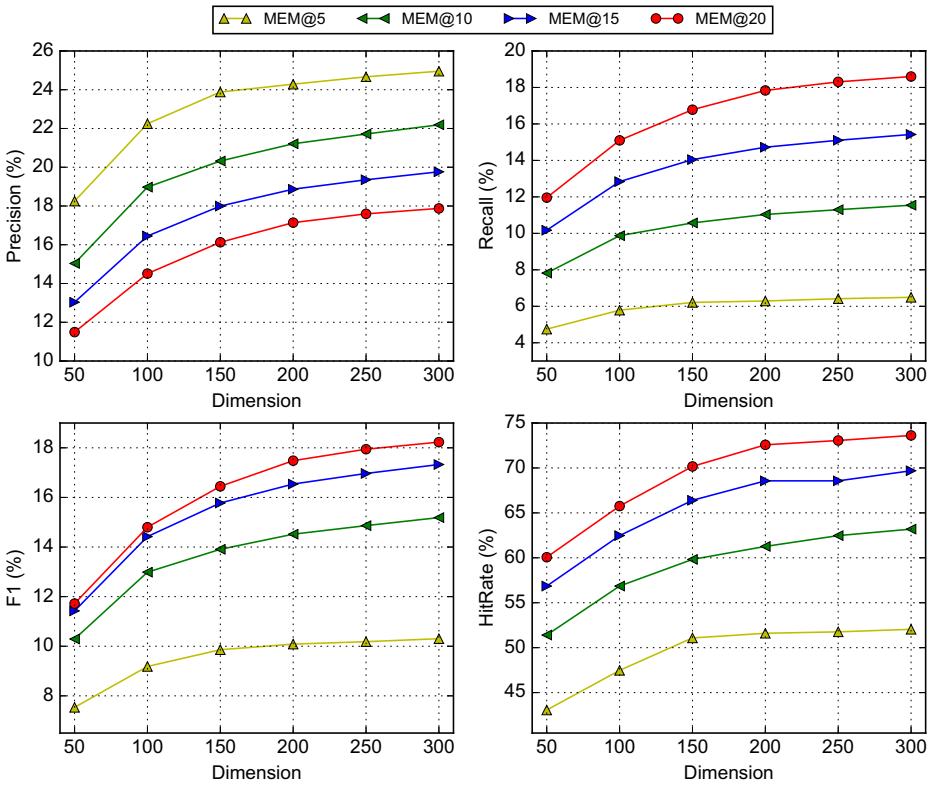

Figure 6 Experimental results of the dimension's impact

We have the following two observations from the experimental results. Firstly, as the dimension increases, the proposed method achieves better performance in terms of precision, recall, F1 score, and hitrate. The reason is that embedding with larger dimension can indeed capture more useful features and depict users and music pieces better. Secondly, the performance tends to be stable when the dimension gets very high. Besides, as shown in Table 10, the approach with high dimension needs more computation cost, which will result in efficiency problem. Finally, we set the dimension of embedding as 200 based on our experiments.

Table 10 The impact of dimension on efficiency

Dimension	Training Time(s)	Testing Time(s)
50	3016.285	391.332
100	3445.858	502.099
150	4228.754	573.51
200	4802.975	705.437
250	5129.111	791.369
300	5604.439	904.171

5.6 Comparison with baselines

We further compare our methods with five state-of-the-art baseline methods, including Bayesian Personalized Ranking (BPR), FISMauc (FISM), Temporal Recommendation based on Injected Preferences Fusion (IPF), Factorizing Personalized Markov Chains (FPMC), and user-based collaborative filtering method (UserKNN). The results are shown in Figure 7.

We have the following observations from the experimental results. (1) Our method has the best performance. Take the F1 score as an example. When compared with BPR, FISM, IPF, FPMC, and UserKNN with the recommending number n=20, the relative performance improvements by MEM are around 96.4%, 69.7%, 42.6%, 31.5%, and 124.1%, respectively. The improvements show that our approach is more effective in contextual preferences inferring and context-aware music recommendation. Besides, it can also be indicated that the users' contextual preferences play an important role in predicting their musical interests and recommending appropriate music. Especially, the proposed approach is better than FPMC because our approach can capture more co-occurrence information instead of only adjacent relation in the sequences, and fully exploit listening sequences, user-music interaction matrix, and metadata.(2) IPF performs better than BPR, FISM, and UserKNN, but is not as good as our method.The reason is that our methods can fully utilize playing sequences and metadata as well as incorporate contextual information in a more effective way. Besides,

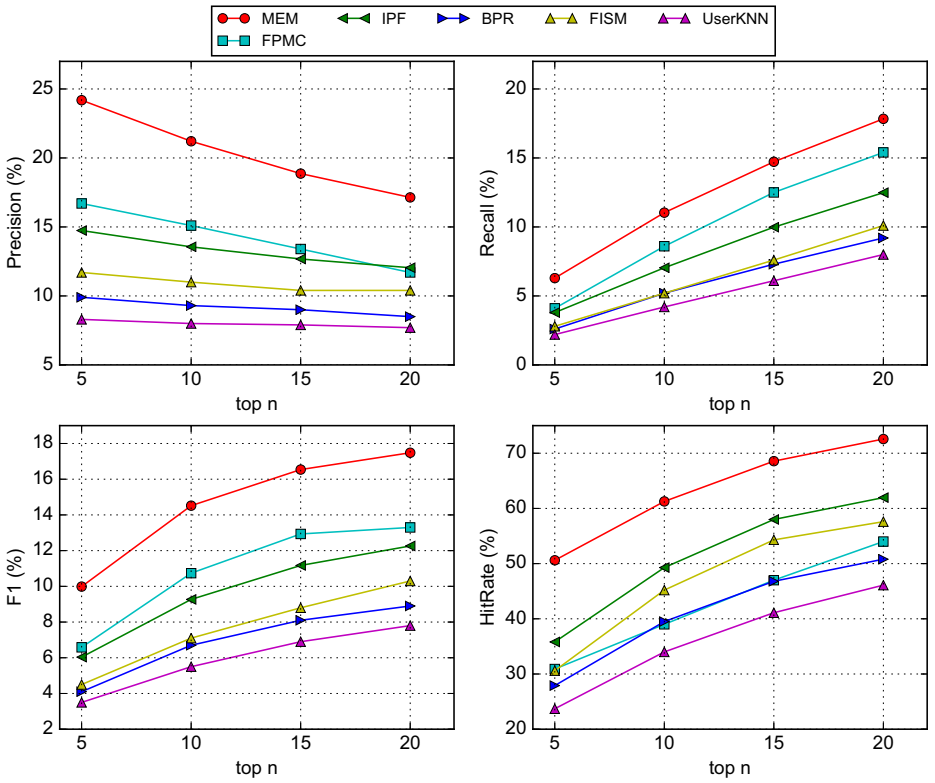

Figure 7 Experimental results of the comparison with baselines

the high sparsity of this dataset (99.72%) may result in the bad performance of the baseline methods. Therefore, we further compare the proposed approach and other baseline methods on datasets with different sparsities in the next subsection. (3) The hitrate and recall for all the three strategies increase but the precision decreases when n gets larger. These results are in accordance with the intuitive and common sense. It requires system developer to select the proper n in order to balance the performances of hitrate/recall and precision.

In conclusion, users' contextual preferences can indeed improve the performance of users' musical interests prediction and music recommendation. It also proves that our method can effectively learn music pieces' embeddings as well as incorporate both the user's global preferences and contextual preferences into music recommendation to satisfy the user's real-time requirements.

5.7 The impact of data sparsity

In order to investigate the proposed method's ability of handling sparse data, we further evaluate our method and the baseline methods on datasets with different sparsities. In this work, the data sparsity means how sparse the user-music interaction data is. Specifically, the datasets with different sparsities are generated by removing music pieces that have been played less than k_m times, where k_m are set to {0,5,10,15,20}. The results are shown in Figure 8.

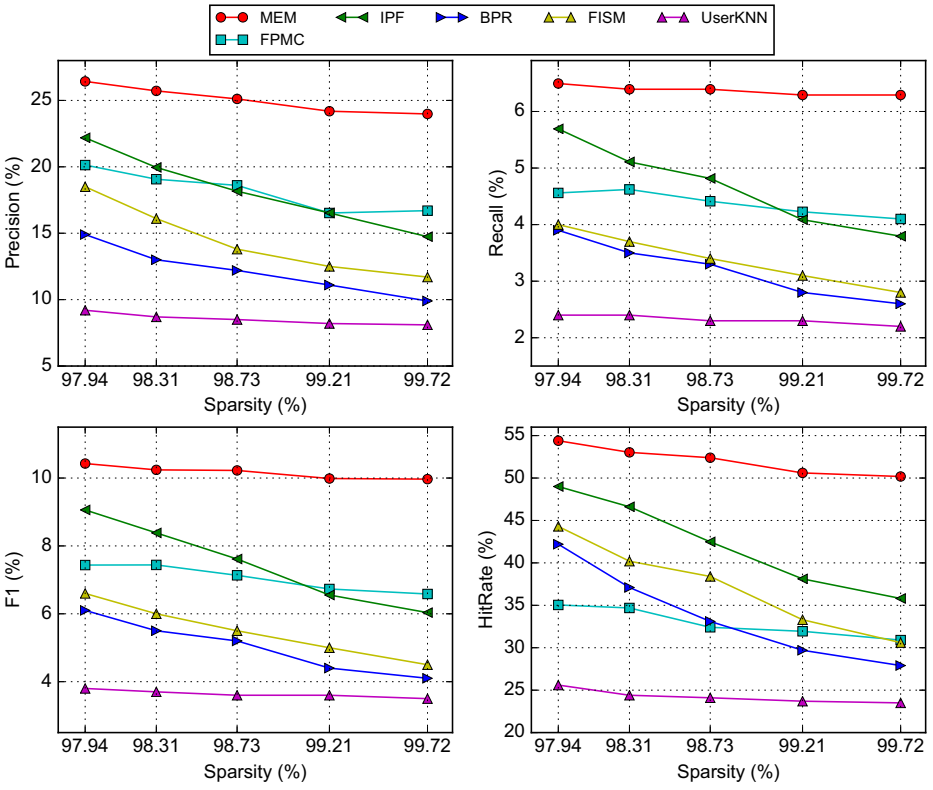

Figure 8 Top 5 Performance over datasets with different sparsities

From the results, we have the following conclusions. (1) Our method has better performance than baseline methods over all datasets with different sparsities. Take the F1 score as an example. When compared with BPR, FISM, IPF, FPMC, and UserKNN with the sparsity set as 97.94%, the relative performance improvements by MEM are around 70.9%, 58.0%, 15.1%, 40.22%, and 174.4%, respectively. This result proves our method can infer, model, and incorporate users' global and contextual musical preferences in a more effective way. Besides, it also verifies the importance of users' global and contextual preferences, especially the latter, in the task of music recommendation. (2) With the sparsity increasing, the performance of all methods, especially BPR and FISM, show obvious trend of decrease. However, the performance gaps between baseline methods and MEM also get larger. Again, take the F1 score as an example. When compared with BPR, FISM, IPF, FPMC, and UserKNN with the sparsity being 99.72%, the relative performance improvements by MEM are around 143.2%, 121.5%, 65.2%, 51.4%, and 184.8%, respectively. This is because MEM depends on both listening sequences and user-item matrix as well as metadata to perform recommendation, and it is less sensitive to the sparsity of user-item dataset. In brief, our method can handle sparse data better than baseline methods.

6 Conclusion and future works

This paper presents a novel approach for context-aware music recommendation, which can learn the embeddings of music pieces, obtain the users' global and contextual preferences for music, and recommend appropriate music pieces that are in accordance with the users' preferences. Specifically, the proposed approach consists of three steps. Firstly, it learns music pieces' embeddings (feature vectors in low-dimension continuous space) from music listening records and corresponding metadata. Then it infers and models users' global and contextual preferences for music from their listening records with the learned embeddings. Finally, it recommends appropriate music pieces according to the target user's preferences to satisfy her/his real-time requirements. Experimental evaluations on a real-world dataset show that the proposed approach outperforms baseline methods, especially on sparse datasets.

Our work differs from prior works in two aspects: (1) the proposed approach depends on listening sequences, metadata, and user-item matrix to perform recommendation, and it is less sensitive to the sparsity of user-item dataset; (2) the proposed approach incorporates both users' global and contextual musical preferences into recommendation, which makes it perform better than baseline methods.

Based on our current work, there are three possible future directions. First, we are going to connect microblog service (such as Twitter) with music service websites (Such as Xiami, Last.fm) to incorporate social relationships into music recommendation [36], and adopt more advanced techniques [37] to further improve the performance. Secondly, this work focuses on music recommendation for individual users, and we will explore the possibility to apply our approach in music recommendation for group users, such as families or parties. Finally, we only evaluate our approach by offline experiments in this work, and we will explore if the users' satisfaction increases when the users listen to the recommended music by online experiments.

Acknowledgements This research work was partially supported by Key Research and Development Project of Zhejiang Province (No. 2015C01027 and No. 2017C01015), National Science Foundation of

China (No. 61772461), Natural Science Foundation of Zhejiang Province (No. LR18F020003 and No. LY17F020014), and Australian Research Council (ARC) Linkage Project under No. LP140100937.

References

1. Adamic, L.A., Huberman, B.A.: Power-law distribution of the World Wide Web. Science **287**(5461), 2115–2115 (2000)
2. Adomavicius, G., Tuzhilin, A.: Context-Aware recommender systems. In: Recommender systems handbook, pp. 217–253. Springer, Boston (2011)
3. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM Press, New York (1999)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
5. Bengio, Y., Schwenk, H., Senécal, J.S., Morin, F., Gauvain, J.L.: Neural probabilistic language models. In: Innovations in machine learning, pp. 137–186. Springer, Berlin (2006)
6. Cai, R., Zhang, C., Wang, C., Zhang, L., Ma, W.Y.: Musicsense: contextual music recommendation using emotional allocation modeling. In: Proceedings of the 15Th international conference on multimedia, pp. 553–556. ACM, New York (2007)
7. Celma, O.: Music recommendation. Springer, Berlin (2010)
8. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on Top-N recommendation tasks. In: Proceedings of the Fourth ACM conference on recommender systems, pp. 39–46. ACM, New York (2010)
9. Deng, S., Wang, D., Li, X., Xu, G.: Exploring user emotion in microblogs for music recommendation. Expert Syst. Appl. **42**(23), 9284–9293 (2015)
10. Dias, R., Fonseca, M.J.: Improving music recommendation in session-based collaborative filtering by using temporal context. In: 2013 IEEE 25th international conference on tools with artificial intelligence (ICTAI), pp. 783–788. IEEE, Washington (2013)
11. Djuric, N., Wu, H., Radosavljevic, V., Grbovic, M., Bhamidipati, N.: Hierarchical neural language models for joint representation of streaming documents and their content. In: Proceedings of the 24Th international conference on World Wide Web, pp. 248–255. International World Wide Web Conferences Steering Committee (2015)
12. Han, B.J., Rho, S., Jun, S., Hwang, E.: Music emotion classification and context-based music recommendation. Multimedia Tools and Applications **47**(3), 433–460 (2010)
13. Hariri, N., Mobasher, B., Burke, R.: Context-Aware music recommendation based on latenttopic sequential patterns. In: Proceedings of the Sixth ACM conference on recommender systems, pp. 131–138. ACM, New York (2012)
14. Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., Yang, S.: Social contextual recommendation. In: Proceedings of the 21St ACM international conference on information and knowledge management, pp. 45–54. ACM, New York (2012)
15. Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for Top-N recommender systems. In: Proceedings of the 19Th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 659–667, ACM (2013)
16. Kaminskas, M., Ricci, F.: Contextual music information retrieval and recommendation: state of the art and challenges. Computer Science Review **6**(2), 89–119 (2012)
17. Kaminskas, M., Ricci, F., Schedl, M.: Location-aware music recommendation using auto-tagging and hybrid matching. In: Proceedings of the 7Th ACM conference on recommender systems, pp. 17–24. ACM, New York (2013)
18. Knees, P., Schedl, M.: A survey of music similarity and recommendation from music context data. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **10**(1), 2 (2013)
19. Liang, H., Xu, Y., Li, Y., Nayak, R., Tao, X.: Connecting users and items with weighted tags for personalized item recommendations. In: Proceedings of the 21St ACM conference on hypertext and hypermedia, pp. 51–60. ACM, New York (2010)
20. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(2579-2605), 85 (2008)
21. Marujo, L., Ling, W., Ribeiro, R., Gershman, A., Carbonell, J., de Matos, D.M., Neto, J.P.: Exploring events and distributed representations of text in multi-document summarization. Knowl.-Based Syst. **94**, 33–42 (2016)

22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
23. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. In: Aistats, vol. 5, pp. 246–252. Citeseer (2005)
24. North, A., Hargreaves, D.: The social and applied psychology of music. OUP Oxford, Oxford (2008)
25. Oulasvirta, A., Hukkinen, J.P., Schwartz, B.: When more is less: the paradox of choice in search engine use. In: Proceedings of the 32Nd international ACM SIGIR conference on research and development in information retrieval, pp. 516–523. ACM, New York (2009)
26. Park, H.S., Yoo, J.O., Cho, S.B.: A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In: Fuzzy systems and knowledge discovery, pp. 970–979. Springer, Berlin (2006)
27. Pettijohn, T.F. II, Williams, G.M., Carter, T.C.: Music for the seasons: seasonal music preferences in college students. Curr. Psychol. **29**(4), 328–345 (2010)
28. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth conference on uncertainty in artificial intelligence, 452–461. AUAI Press (2009)
29. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19Th international conference on World Wide Web, pp. 811–820. ACM, New York (2010)
30. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work, pp. 175–186. ACM, New York (1994)
31. Rho, S., Han, B.J., Hwang, E.: Svr-based music mood classification and context-based music recommendation. In: Proceedings of the 17Th ACM international conference on multimedia, pp. 713–716. ACM, New York (2009)
32. Schedl, M., Vall, A., Farrahi, K.: User geospatial context for music recommendation in microblogs. In: Proceedings of the 37Th international ACM SIGIR conference on research and development in information retrieval, pp. 987–990. ACM, New York (2014)
33. Seth, A., Zhang, J., Cohen, R.: A personalized credibility model for recommending messages in social participatory media environments. World Wide Web **18**(1), 111–137 (2015)
34. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: Proceedings of the 38Th International ACM SIGIR conference on research and development in information retrieval, pp. 403–412. ACM, New York (2015)
35. Wang, X., Rosenblum, D., Wang, Y.: Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20Th ACM international conference on multimedia, pp. 99–108. ACM, New York (2012)
36. Wang, Y., Li, L., Liu, G.: Social context-aware trust inference for trust enhancement in social network based recommendations on service providers. World Wide Web **18**(1), 159–184 (2015)
37. Wu, J., Chen, L., Yu, Q., Han, P., Wu, Z.: Trust-aware media recommendation in heterogeneous social networks. World Wide Web **18**(1), 139–157 (2015)
38. Wu, X., Liu, Q., Chen, E., He, L., Lv, J., Cao, C., Hu, G.: Personalized next-song recommendation in online karaokes. In: Proceedings of the 7Th ACM conference on recommender systems, pp. 137–140. ACM, New York (2013)
39. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long-and short-term preference fusion. In: Proceedings of the 16Th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 723–732. ACM, New York (2010)
40. Yao, W., He, J., Huang, G., Cao, J., Zhang, Y.: A graph-based model for context-aware recommendation using implicit feedback data. World Wide Web **18**(5), 1351–1371 (2015)
41. Yu, Z., Zhou, X., Yu, Z., Zhang, D., Chin, C.Y.: An osgi-based infrastructure for context-aware multimedia services. IEEE Commun. Mag. **44**(10), 136–142 (2006)
42. Yu, Z., Zhou, X., Zhang, D., Chin, C.Y., Wang, X., Men, J.: Supporting context-aware media recommendations for smart phones. IEEE Pervasive Comput. **5**(3), 68–75 (2006)
43. Zhou, G., Zhou, Y., He, T., Wu, W.: Learning semantic representation with neural networks for community question answering retrieval. Knowl.-Based Syst. **93**, 75–83 (2016)