Music recommendation via heterogeneous information graph embedding

Conference Paper · May 2017

DOI: 10.1109/IJCNN.2017.7965907

3 authors:

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

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

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

Music Recommendation via Heterogeneous Information Graph Embedding

Dongjing Wang
College of Computer Science and Technology,
Zhejiang University, China;
Advanced Analytics Institute,
University of Technology Sydney, Australia
Email: tokyo1@zju.edu.cn

Guandong Xu
Advanced Analytics Institute,
University of Technology Sydney, Australia
Email: Guandong.Xu@uts.edu.au

Shuiguang Deng
College of Computer Science and Technology,
Zhejiang University, China
Email: dengsg@zju.edu.cn

***Abstract*—Traditional music recommendation techniques suffer from limited performance due to the sparsity of user-music interaction data, which is addressed by incorporating auxiliary information. In this paper, we study the problem of personalized music recommendation that takes different kinds of auxiliary information into consideration. To achieve this goal, a Heterogeneous Information Graph (HIG) is first constructed to encode different kinds of heterogeneous information, including the interactions between users and music pieces, music playing sequences, and the metadata of music pieces. Based on HIG, a Heterogeneous Information Graph Embedding method (HIGE) is proposed to learn the latent low-dimensional representations of music pieces. Then, we further develop a context-aware music recommendation method. Extensive experiments have been conducted on real-world datasets to compare the proposed method with other state-of-the-art recommendation methods. The results demonstrate that the proposed method significantly outperforms those baselines, especially on sparse datasets.**

I. Introduction

Nowadays, there is an enormous amount of digital music available on the Internet. For example, Apple Music currently offers over 30 million pieces of music (http://support.apple.com/en-us/HT204951). Therefore, it is becoming increasingly difficult for people to find the music that they most enjoy, which is known as the Paradox of Choice [1]. Therefore, music recommendation has become an interesting topic in both research and industry. Similar to recommender systems applied in various domains [2]–[4], music recommendation has greatly benefited from the algorithmic advances of the recommender system community, e.g., collaborative filtering (CF). However, traditional CF methods usually suffer from limited performance when user-item interaction data are sparse, which is very common for online music services where the amount of music can easily reach several millions.

Hybrid recommender systems [5], [6] are proposed to address this problem by combining collaborative filtering and auxiliary information such as item content and associated textual descriptions. However, previous studies on hybrid music recommender systems ignore music playing sequence, which is important auxiliary information for music representation learning [7]. In particular, unlike books or products, music is a carrier of thoughts and emotions instead of a neutral item, and the sequences of music liked or played by people often reflect their specific music tastes and preferences during the corresponding period of time. On the other hand, music pieces with similar intrinsic features tend to co-occur in the same music playing sequences.

Motivated by the discussion above, we present a context-aware music recommendation approach, which is able to recommend appropriate music pieces to users. In analogy to matrix factorization methods for collaborative filtering, the proposed approach does not require music pieces to be represented by the features ahead, but it learns the latent representations from the data of users' music playing and the metadata of music. Specifically, a Heterogeneous Information Graph (HIG) is first constructed to encode different kinds of information, including the interactions between users and music pieces, music playing sequences, and the metadata of music pieces. In this way, HIG captures music co-occurrences at the user level, local context level, and metadata level. In particular, this co-occurrence information in HIG indicates the music pieces' intrinsic features, so they can be used to learn the representation of music. Then, based on HIG, a Heterogeneous Information Graph Embedding (HIGE) method is proposed to learn the latent low-dimensional representations (embeddings) of music pieces by considering music co-occurrence information at different levels. In this way, the music pieces that have similar intrinsic features yield similar embeddings. Finally, a context-aware music recommendation approach based on HIGE is proposed to recommend appropriate music pieces to satisfy users' real-time requirements. The main contributions of this paper are summarized as follows:

- We propose HIG to encode different kinds of information, especially music playing sequences, in a unified manner;
- We devise HIGE method to learn the latent low-dimensional representations of music pieces from HIG;
- We propose a context-aware music recommendation method based on HIGE and conduct extensive experiments on real-world datasets. The results show that our method outperforms baseline methods, especially on sparse datasets.

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

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

II. Related Work

In this section, we describe the existing work on personalized music recommendation, as well as the studies on graph embedding that inspired our work.

A. Personalized Music Recommendation

Existing music recommendation techniques are broadly categorized into collaborative-based, content-based, context-aware, and hybrid approaches [8]. Collaborative-based methods [9] estimate the similarity between users based on their listening records and recommend music by referring to the preferences of similar users. Content-based methods [10], [11] compute the similarity between music pieces based on music content or associated textual descriptions, and recommend music pieces that are similar to those the users liked in the past. Context-aware approaches [12]–[14] are based on the fact that the environment or the users' state of mind may influence their music preferences. Hybrid methods [5], [6] combine the techniques from those three basic approaches to address the data sparsity problem and usually achieve better recommendation performance. To the best of our knowledge, existing hybrid music recommender systems mainly focus on social relationships or content. Our work discusses leveraging different kinds of information, especially music playing sequences, in a heterogeneous information graph for better music recommendation. In addition, regarding data characteristics, most existing work handles users' explicit feedback such as item ratings. Recently, there has been increasing attention on the usage of users' implicit feedback to conduct collaborative filtering [15]–[17], which is much easier to collect.

B. Graph Embedding

Our work is inspired by classical methods of graph embedding or dimension reduction in general, such as DeepWalk [18], RANGE [19], LINE [20], GraRep [21], and TADW [22]. Specifically, Perozzi et al. [18] propose an approach named DeepWalk, which deploys a truncated random walk for social network embedding. In DeepWalk, nodes with similar neighbors yield similar representations. Yang et al. [22] prove that DeepWalk is actually equivalent to matrix factorization, and also propose text-associated DeepWalk (TADW), which incorporates the text features of vertices into network representation learning under the framework of matrix factorization. Tang et al. [20] present a novel network embedding model called LINE, which utilizes both edge weights and neighbors to learn the embeddings of nodes. Those powerful, efficient models have obtained promising results on many tasks. Recently, the concept of graph embedding has been expanded to many research fields, such as dimension reduction [23], question answering [24], recommendation [25], and object tracking [26].

III. Problem Definition

For ease of presentation, we define the key data structures and notations used in this paper. Table I also lists them.

TABLE I
Notations used in this paper

Variable	Interpretation
U, M	the set of users and music pieces
H^u	the historical music playing sequence of user $u \in U$
D	the metadata set
G, E, w, r	the graph, edge set, the direct edge weight and the indirect relationship weight
$\vec{v}_m$	the embedding of music piece $m \in M$
$\vec{p}_g^u$, $\vec{p}_c^u$	user u's global and contextual music preferences

Let $U = \{u_1, u_2, ..., u_{|U|}\}$ be the user set and $M = \{m_1, m_2, ..., m_{|M|}\}$ represent the music set. For each user $u \in U$, her/his historical music playing sequence is a list of music records, which is formally defined as $H^u = \{m_1^u, m_2^u, ..., m_{|H^u|}^u\}$. As shown in Figure 1, each music record $m_i^u \in M$ in H^u is ordered according to its playing time. Therefore, the task becomes recommending music pieces that u probably enjoys currently given her/his historical music playing sequence H^u.

In order to learn the unified latent representations (embeddings) of music from the given music playing data and the metadata, we first propose a Heterogeneous Information Graph (HIG), which incorporates the users-music interaction information, music playing sequences, and the metadata of music in a unified manner. Specifically, as shown in Figure 1, HIG consists of three components: user-music interaction graph, music-music transition graph, and music-metadata knowledge graph.

Definition 1. (**User-Music Interaction Graph**) The user-music interaction graph, denoted as $G_{u,m} = (U \cup M, E_{u,m})$, includes the interaction information between users and music pieces in the music playing data. $E_{u,m}$ is the set of edges between users and music pieces. Besides, the original edge weights range from 0 to ∞, so we adopt hyperbolic tangent sigmoid function to normalize them into $[0, 1)$. Specifically, the weight of the edge between user u_i and music m_j is defined as $w_{u_i,m_j} = tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$, where t is the number of times that u_i has listened to m_j.

Obviously, the user-music interaction graph encodes music co-occurrences at the user level, which is equivalent to a user-music interaction matrix that is widely explored in traditional collaborative filtering approaches. In particular, the user-music graph is constructed based on the idea that each user usually has specific preferences and the music pieces with similar intrinsic features are more likely to be listened to by the same users. Beyond the user level information, we also introduce

Music Playing Data and Metadata

User	Record	Music	Playing Time	Singer	Album	Tag
u_1	$m_1^{u_1}$	My Heart Will Go On	2015-09-23 19:52	Celine Dion	Let's Talk About Love	pop, female vocalists, love, soundtrack
u_1	$m_2^{u_1}$	I Do It for You	2015-09-23 19:56	Bryan Adams	So Far So Good	rock, ballad, classic rock, pop
u_1	$m_3^{u_1}$	Thank You For Loving Me	2015-09-23 20:03	Bon Jovi	Crush	rock, classic rock, ballad, hard rock
u_1	$m_4^{u_1}$	It's My Life	2015-09-23 20:08	Bon Jovi	Crush	rock, classic rock, hard rock
u_2	$m_1^{u_2}$	Enter Sandman	2015-09-24 10:49	Metallica	The Metallica Collection	rock, metal, heavy metal, thrash metal
u_2	$m_2^{u_2}$	Nothing Else Matters	2015-09-24 10:54	Metallica	The Metallica Collection	rock, metal, heavy metal, ballad
u_2	$m_3^{u_2}$	It's My Life	2015-09-24 11:01	Bon Jovi	Crush	rock, classic rock, hard rock

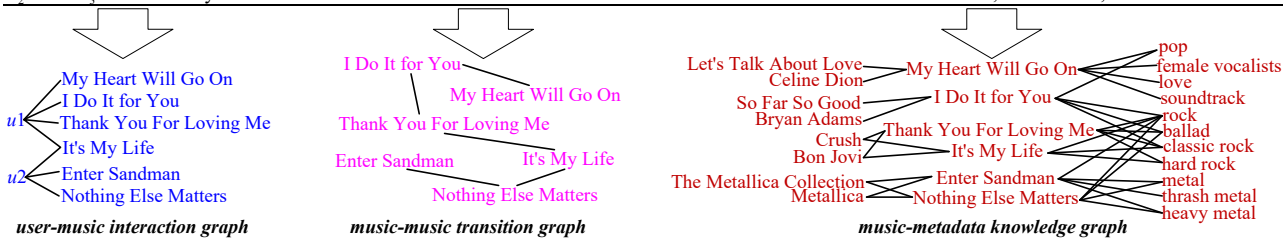


Fig. 1. Illustration of constructing a heterogeneous information graph from music playing data and the metadata

the music-music transition graph to capture the music co-occurrences in local contexts.

Definition 2. (**Music-Music Transition Graph**) The music-music transition graph, denoted as $G_{m,m} = (M, E_{m,m})$, incorporates the music playing sequence information in the music playing data. $E_{m,m}$ is the set of edges between music pieces. The weight of the edge between music m_i and m_j is defined using hyperbolic tangent sigmoid function as $w_{m_i,m_j} = tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$, where t is the number of times that m_i and m_j co-occur in the context window of music playing sequence, whose size is set to 2 in the example given in Figure 1.

The music-music transition graph is introduced based on the fact that music pieces with similar intrinsic features tend to co-occur in the same music playing sequences, which reflect users' specific preferences for music during the corresponding period of time. Specifically, we do not incorporate the playing order information based on the following two reasons. First, the music co-occurrence information is adequate for the proposed embedding model, and incorporating the playing order information will increase the complexity of the algorithm but not necessarily increase the performance. Second, the playing order information is more sparse than co-occurrence information especially on sparse dataset, which will decrease the performance of the proposed approach.

Generally, the user-music graph and music-music graph encode the behavior information in the music playing data, and capture the music co-occurrences at both the user level and the local context level. In addition, music pieces with similar intrinsic features tend to have similar metadata. Therefore, we further introduce the music-metadata knowledge graph to capture music co-occurrences at the metadata level.

Definition 3. (**Music-Metadata Knowledge Graph**) The music-metadata knowledge graph, denoted as $G_{m,d} = (M \cup D, E_{m,d})$, encodes the metadata of the music pieces, including singers, albums, and tags. D is the metadata set, and $E_{m,d}$ is the set of edges between music pieces and metadata. The weight of the edge between music m_i and metadata d_j is defined using hyperbolic tangent sigmoid function as $w_{m_i,d_j} = tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$, where t is the number of times that m_i is assigned with d_j. These three types of graphs defined above are further integrated into one heterogeneous information graph.

Definition 4. (**Heterogeneous Information Graph, HIG**) The heterogeneous information graph is the combination of music-music, music-user, and music-metadata graphs constructed from the music playing data and the metadata of all music pieces. It captures music co-occurrences at different levels. Note that the HIG can be generalized to integrate other types of information, such as the music-playlist association graph and music-audio similarity graph. In the HIG, we mainly focus on learning the embeddings of music pieces. The representations of the users' preferences for music can then be computed by aggregating the embeddings of music.

Finally, we formally define the heterogeneous information graph embedding method as follows:

Definition 5. (**Heterogeneous Information Graph Embedding, HIGE**) Given a large collection of music playing data and metadata, HIGE aims to learn the latent low-dimensional representations of music pieces by embedding the heterogeneous information graph constructed from the collection into a low-dimensional embedding space.

IV. Proposed Approach

The proposed approach consists of two components: heterogeneous information graph embedding and context-aware music recommendation.

A. Heterogeneous Information Graph Embedding

The Heterogeneous Information Graph Embedding (HIGE) method is proposed to learn the embeddings of music from the HIG. Here, the key idea behind HIGE is that the co-occurrence information indicates the similarity or relevance between music pieces. In other words, music pieces with

similar intrinsic features tend to co-occur at the local context level, user level, or metadata level, and should be represented closely in the low-dimensional embedding space.

As for two music vertices m_i and m_j in HIG, we define the joint probability between m_i and m_j as follows:

$$p\left(m_i, m_j\right) = \frac{\exp\left(\vec{v}_{m_i}^T \cdot \vec{v}_{m_j}\right)}{\sum\limits_{m_k \in M} \exp\left(\vec{v}_{m_k}^T \cdot \vec{v}_{m_j}\right)}, \tag{1}$$

where $\vec{v}_{m_i}$ and $\vec{v}_{m_j}$ are the embeddings of music pieces m_i and m_j. Equation 1 defines a distribution $p\left(\cdot,\cdot\right)$ over the space $M \times M$. To preserve the structure and information of the heterogeneous information graph, we make the joint distribution $p\left(\cdot,\cdot\right)$ be close to its empirical distribution $\hat{p}\left(\cdot,\cdot\right)$, which can be achieved by minimizing the following objective function:

$$O = d\left(\hat{p}\left(\cdot,\cdot\right), p\left(\cdot,\cdot\right)\right), \tag{2}$$

where $d\left(\cdot,\cdot\right)$ is the Kullback-Leibler divergence between two distributions.

As mentioned above, music pieces with similar intrinsic features tend to have similar co-occurrence information, and should yield similar embeddings. Therefore, the empirical distribution of two music pieces consists of two parts: the direct edge weight in the music-music transition graph and the indirect relationship weight in the user-music interaction graph and the music-metadata knowledge graph, which correspond to the music co-occurrences at the local context level and the music co-occurrences at the user and metadata level, respectively. Formally, the empirical distribution is defined as

$$\hat{p}(m_i, m_j) = \frac{w_{m_i,m_j} + r_{m_i,m_j}}{\sum\limits_{m_k \in M} \left(w_{m_k,m_j} + r_{m_k,m_j}\right)}, \tag{3}$$

where w_{m_i,m_j} is the direct edge weight in the music-music transition graph, and r_{m_i,m_j} is the indirect relationship weight in the user-music interaction graph and the music-metadata knowledge graph, which is defined as

$$r_{m_i,m_j} = \cos\left(\vec{n}_{G_{u,m}}^{m_i} \oplus \vec{n}_{G_{m,d}}^{m_i},\ \vec{n}_{G_{u,m}}^{m_j} \oplus \vec{n}_{G_{m,d}}^{m_j}\right), \tag{4}$$

where $\vec{n}_{G_{u,m}}^{m_i} = \left(w_{m_i,u_1}, ..., w_{m_i,u_{|U|}}\right)$ is the neighbor weight vector of m_i in the user-music interaction graph, $\vec{n}_{G_{m,d}}^{m_i} = \left(w_{m_i,d_1}, ..., w_{m_i,d_{|D|}}\right)$ is the neighbor weight vector of m_i in the music-metadata knowledge graph, and $\cos\left(\cdot,\cdot\right)$ is the cosine similarity of two vectors.

By replacing the distance function with KL-divergence and omitting some constants, we get the final objective function as follows:

$$\begin{aligned} O &= d\left(\hat{p}(\cdot,\cdot), p(\cdot,\cdot)\right) \\ &= \sum_{m_j \in M} \sum_{m_i \in M} d\left(\hat{p}(m_i, m_j), p(m_i, m_j)\right) \\ &= \sum_{m_j \in M} \sum_{m_i \in M} \hat{p}(m_i, m_j) \log \frac{\hat{p}(m_i,m_j)}{p(m_i,m_j)} \\ &\propto -\sum_{E_{m,m}} \left(w_{m_i,m_j} + r_{m_i,m_j}\right) \log p(m_i, m_j). \end{aligned} \tag{5}$$

Our goal is to learn the embeddings of music pieces by minimizing the objective function.

Learning. In the learning phase, we need to minimize the objective function of the log probability defined in Equation 5 over the whole heterogeneous information graph. However, the complexity of computing the corresponding soft-max function defined in Equation 1 is proportional to the music set size $|M|$, which can easily reach several millions. Therefore, it is difficult to directly compute the probability. In this paper, we adopt negative sampling [27] to approximate the soft-max function defined in Equation 1. For each edge (m_i, m_j), it specifies the following objective function:

$$\begin{aligned} &\log\sigma\left(\vec{v}_{m_i}^T \cdot \vec{v}_{m_j}\right) \\ &\quad + k \cdot E_{m_n \sim P(m)}\left[\log\sigma\left(-\vec{v}_{m_n}^T \cdot \vec{v}_{m_j}\right)\right], \end{aligned} \tag{6}$$

where $\sigma(x) = 1/(1+e^{-x})$, and k is the number of negative samples. $E_{m_n \sim P(m)}\left[\cdot\right]$ is the expectation value with respect to the noise distribution $P(m)$, which is modeled by empirical unigram distribution over all music pieces. The negative sampling method generates k noise samples for prediction, where $k \ll |M|$. Then, we adopt the stochastic gradient algorithm for optimizing Equation 6. In each step, a binary edge (m_i, m_j) is sampled with the probability proportional to its weight w_{m_i,m_j}, and meanwhile multiple negative edges (m_n, m_j) are sampled from a noise distribution $P(m)$. Specifically, for a sampled edge (m_i, m_j), the gradient with respect to the embedding vector $\vec{v}_{m_i}$ of vertex m_i will be calculated as:

$$\frac{\partial O}{\partial \vec{v}_{m_i}} = \left(w_{m_i,m_j} + r_{m_i,m_j}\right) \cdot \frac{\partial \log p\left(m_i, m_j\right)}{\partial \vec{v}_{m_i}} \tag{7}$$

We can see that the gradient is multiplied by the weight of the edge. It is very hard to find a good learning rate especially when the weights of edges have a high variance, and inappropriate learning rate will cause gradient explosion or vanishing. To overcome this problem, we adopt the edge sampling approach used in [20]. Let $W = \left(w_1 + r_1, w_2 + r_2, ..., w_{|E|} + r_{|E|}\right)$ denote the sequence of edge weights. First, we calculate the sum of all weights $w_{sum} = \sum_{i=1}^{|E|} w_i + r_i$. Then, sample a random value within $[0, w_{sum}]$ to see which interval $[\sum_{j=0}^{i-1} w_j + r_j, \sum_{j=0}^{i} w_j + r_j)$ the value falls into. This approach takes $O(|E|)$ time to draw a sample, which is computationally expensive when the number of edges $|E|$ is large. We use the alias table method proposed in [28] to draw a sample according to the weights of the edges, which takes only $O(1)$ time when repeatedly drawing samples from the same discrete distribution. Moreover, optimization with negative sampling takes $O(d \times (k+1))$ time, where k is the number of negative samples and d is the time taking for one sampling. Therefore, the entire step takes $O(d \times k)$ time. Actually, the number of steps used for optimization is usually proportional to the number of edges $|E|$, so the overall time complexity of optimization is $O(d \times k \times |E|)$. d and k are all constants, so the final complexity is linear to the number of edges $|E|$, and does not depend on the number of vertices.

Finally, the embedding of each music piece is learned by HIGE efficiently, and similar music pieces lie nearby in the low-dimensional embedding space.

B. Context-aware Music Recommendation

Based on these learned embeddings, we can infer and model the users' global and contextual preferences from their music playing sequences [12], [29]. Specifically, the user u's global preference is reflected in u's music playing histories, and it can be obtained by averaging the embeddings of music pieces in her/his historical music playing sequence H^u, which is formally defined as

$$\vec{p}_g^{\,u} = \frac{1}{|H^u|} \sum_{1 \le i \le |H^u|} \vec{v}_{m_i^u}. \tag{8}$$

Similarly, u's contextual music preference is reflected in u's recent music playing histories, and it can be obtained by averaging the embeddings of music pieces in her/his recent music playing sequence (music pieces in her/his active interaction session), which is defined as

$$\vec{p}_c^{\,u} = \frac{1}{|S_u|} \sum_{m_i^u \in S_u} \vec{v}_{m_i^u}, \tag{9}$$

where S_u is u's recent music playing sequence in her/his current interaction with the system.

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

$$p\left(m_i|\vec{p}_g^{\,u}, \vec{p}_c^{\,u}\right) = p\left(m_i|\vec{p}_g^{\,u}\right) + p\left(m_i|\vec{p}_c^{\,u}\right), \tag{10}$$

where $p\left(m_i|\vec{p}_g^{\,u}\right)$ is the predicted global preference of user u for music piece m_i and $p\left(m_i|\vec{p}_c^{\,u}\right)$ is the predicted contextual preference of u for m_i. $p\left(m_i|\vec{p}_g^{\,u}\right)$ and $p\left(m_i|\vec{p}_c^{\,u}\right)$ are formally defined via a soft-max function as follows:

$$p(m_i|\vec{p}_g^{\,u}) = \frac{\exp\left(\vec{v}_{m_i}^T \cdot \vec{p}_g^{\,u}\right)}{\sum_{m_j \in M} \exp\left(\vec{v}_{m_j}^T \cdot \vec{p}_g^{\,u}\right)}, \tag{11}$$

$$p(m_i|\vec{p}_c^{\,u}) = \frac{\exp\left(\vec{v}_{m_i}^T \cdot \vec{p}_c^{\,u}\right)}{\sum_{m_j \in M} \exp\left(\vec{v}_{m_j}^T \cdot \vec{p}_c^{\,u}\right)}, \tag{12}$$

where $\vec{v}_{m_i}$ and $\vec{v}_{m_j}$ is the learned embedding of music piece m_i and m_j.

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

$$m_i >_{\vec{p}_g^{\,u},\vec{p}_c^{\,u}} m_j :\Leftrightarrow p\left(m_i|\vec{p}_g^{\,u}, \vec{p}_c^{\,u}\right) > p\left(m_j|\vec{p}_g^{\,u}, \vec{p}_c^{\,u}\right). \tag{13}$$

Finally, we can recommend the music pieces with high ranking scores to the target user.

V. Experiments

In this section, we evaluate the following: (1) what do the learned embeddings look like; (2) how does the dimension of the embeddings affect the recommendation performance; (3) how does the proposed approach perform when compared with other state-of-the-art recommendation techniques; and (4) how do the proposed method and the baseline methods perform on datasets with different sparsities.

A. Dataset

To evaluate the proposed approach, we collect a real-world dataset from an online music service website named Xiami Music (http://www.xiami.com, the Chinese version of Last.fm). As shown in Table II, the dataset contains 4,284,000 interaction records between 4,284 users and 361,861 music pieces. In addition, Figure 2 illustrates the relationship between popularity (listening count) and the amount of music pieces with corresponding popularity. We can see that, only a small amount of music pieces are very popular, while the majority of music are not so popular, which basically conforms to the power law distribution [30].

TABLE II
Complete statistics of the dataset

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

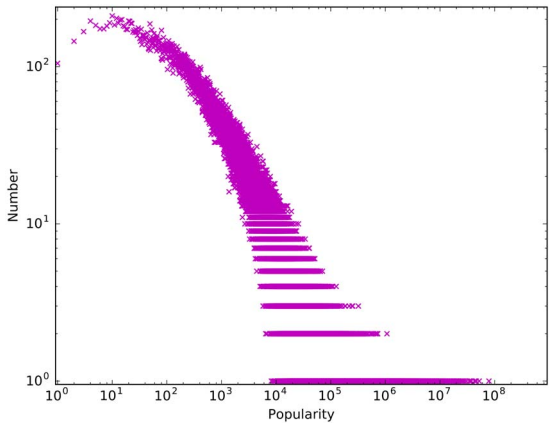


Fig. 2. Popularity analysis of the dataset

B. Baseline Methods

In last two decades, many algorithms have been proposed for top-n recommendation on binary data without rating. Four state-of-the-art recommendation approaches, namely temporal recommendation based on Injected Preferences Fusion (IPF) [31], Bayesian Personalized Ranking (BPR) [16], FISMauc (FISM) [32], and the user-based collaborative filtering method (UserKNN) [33], are used as baseline methods.

C. Experimental Design and Evaluation Metrics

In order to evaluate the performance of different recommendation methods, we split the whole dataset into training sets and test sets according to the idea of 5-fold cross-validation, and the window size in the music-music transition graph is set to 7 empirically. In each validation, we keep the complete listening sequences of 80% users and half of the remaining 20% users historical listening sequences as the training set, and use the following half of the remaining 20% users records

as the test set. For each recommendation, we generate a list of n music pieces, denoted by R. The following four metrics [34] are used to evaluate all the recommendation approaches.

Precision, Recall, and F1 Score. Precision (also called positive predictive value) is the fraction of recommended music pieces that are relevant, and recall (also known as sensitivity) is the fraction of relevant music pieces that are recommended. F1 score is the harmonic mean of precision and recall. Formally, the definitions are given as follows:

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

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

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall},$$

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

Hitrate. Hitrate is the fraction of hits, where a hit means the recommendation list contains at least one piece of music that the user actually listened to. The definition is given as

$$HitRate = \frac{\#(hits)}{\#(recs)},$$

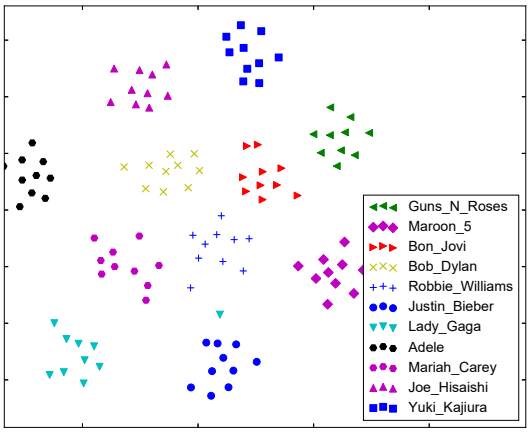
where $\#(hits)$ is the total number of hits and $\#(recs)$ is the total number of recommendation.

D. Illustration of Embedding

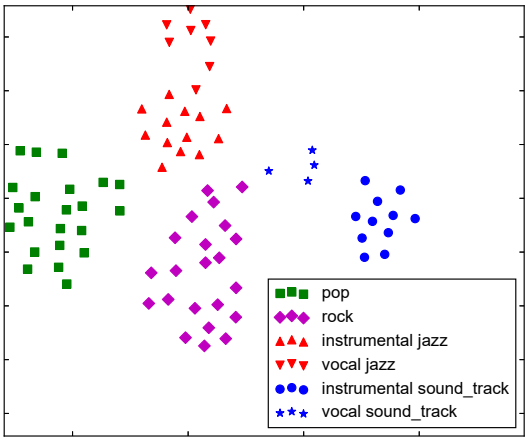
Before starting the quantitative evaluation, we first give a qualitative illustration of the learned embeddings. As shown in Figure 3, the music pieces of the same/similar artists cluster tightly, and the music pieces with similar genres also lie nearby in the 2-dimensional space. The results show that the embeddings learned by our method from music co-occurrence information at different levels depict the intrinsic features of music pieces effectively. Furthermore, some slight differences in genres are also reflected in the learned embeddings, which further demonstrates the effectiveness of our method. For example, as shown in Figure 3(b), vocal jazz music pieces and instrumental jazz music pieces form two close clusters instead of one cluster. On the other hand, the illustration also shows that the learned embeddings are useful for many other tasks, such as similarity measure, corpus visualization, automatic tagging, and classification.

E. Impact of Dimension

The dimension of the embeddings plays an important role in balancing the performance in terms of accuracy and efficiency. Specifically, the embeddings of higher dimension can capture more useful features and depict music pieces better at the cost of lower efficiency. Therefore, to investigate how the dimension of embeddings affects the performance, we first evaluate the proposed approach with different dimensions (50,100,150,200,250,300), and the results are shown in Figure 4. We can see that as the dimension increases, the proposed method achieves better performance in all four metrics. The reason for this is that embedding with a larger dimension can indeed capture more useful features. In addition, the performance tends to be stable when the dimension becomes larger and larger. Finally, the dimension of the embeddings is set to 300 in consideration of accuracy and efficiency.



(a) singer



(b) genre

Fig. 3. The illustration of the learned embeddings in 2-dimentsional space with t-sne [35]

F. Comparison with Baselines

We further compare our method with four state-of-the-art baseline methods, and the results are shown in Figure 5. We can see that our method has the best performance. Take the F1 results of $n = 20$ as an example. When compared with BPR, FISM, IPF, and UserKNN, the relative performance improvements achieved by HIGE are around 104.3%, 76.6%, 48.3%, and 133.14%, respectively. The reason for this is three-fold. First, HIGE incorporates different kinds of heterogeneous information, especially the music playing sequences, while the baselines do not, which also shows that the music playing sequences are indeed useful for learning music embedding and music recommendation. Second, HIGE considers the users' contextual preferences while the baselines except for IPF

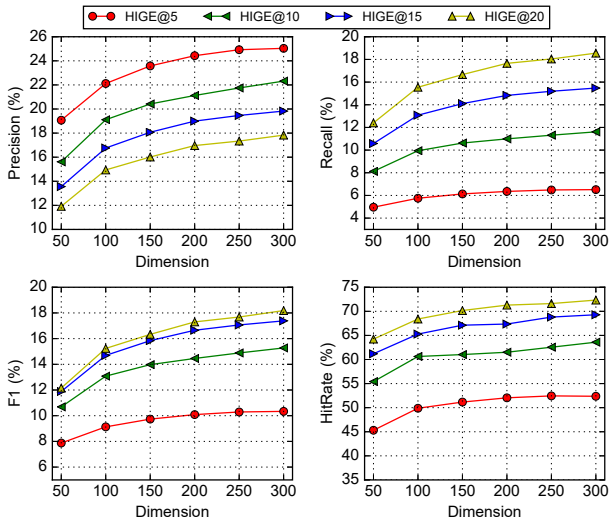


Fig. 4. Experimental results of the dimension's impact

do not, which indicates that users' contextual preferences play an important role in predicting their preferences for music and recommending appropriate music. Third, the high sparsity of the user-music data (99.72%) may also influence the performance of the baseline methods, which is further explored in the next subsection. In conclusion, the proposed method can effectively learn music pieces' embeddings as well as make better recommendation.

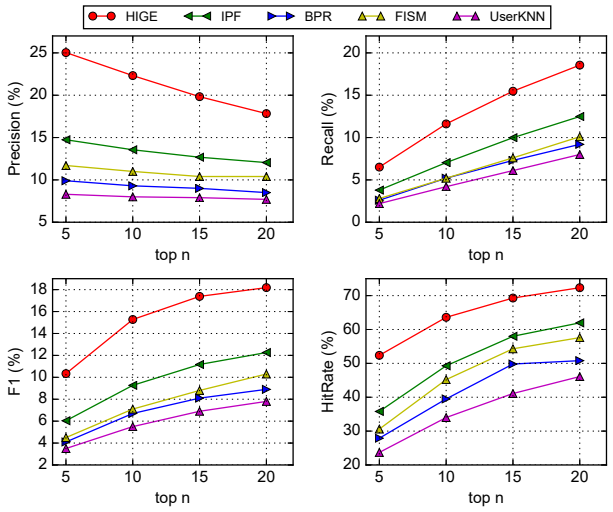


Fig. 5. Performance comparison with baselines

G. Impact of Data Sparsity

To investigate the proposed method's ability to handle sparse data, we further evaluate our method and the baseline methods on datasets with different sparsities. From the results shown in Figure 6, we can see that our method still has the best performance over all datasets. Take the F1 results of sparsity=97.94% as an example. When compared with BPR, FISM, IPF, and UserKNN, the relative performance improvements achieved by HIGE are around 138.7%, 120.6%, 60.7%, and 183.0%, respectively. In addition, with sparsity increasing, the performance gaps between HIGE and the baseline methods increase to 152.1%, 129.7%, 71.3%, and 209.1%, respectively. This is because HIGE depends on three kinds of information, especially the music playing sequences, to perform recommendation, and it is less sensitive to the sparsity of user-music data. In brief, our method can handle sparse data better than baseline methods.

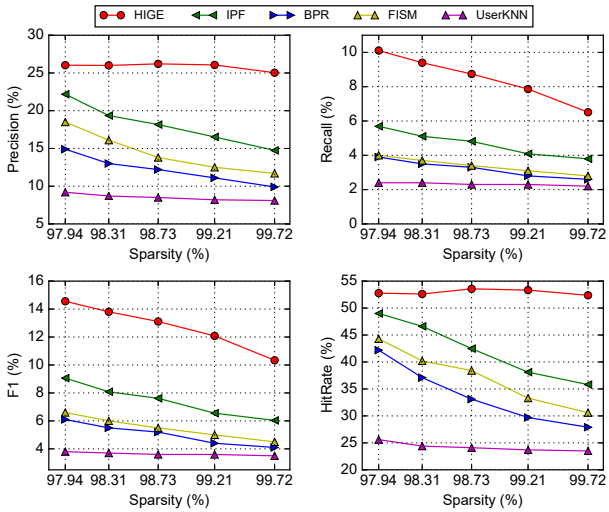


Fig. 6. Performance of n=5 over datasets with different sparsities

VI. Conclusion and Future Work

This paper presents a context-aware music recommendation approach based on heterogeneous information graph embedding, which is able to recommend appropriate music pieces to users. Our work differs from prior work in that the proposed approach incorporates music co-occurrence information at different levels, including the user level, local context level, and metadata level, to learn the embeddings of music and perform recommendation. Extensive experiments have been conducted on real-world datasets, and the results show the effectiveness of our approach, especially on sparse datasets. Based on our current work, we plan to adopt advanced recommending techniques, such as deep learning [36], [37], to further improve the performance. In addition, we will explore if users' satisfaction can be increased by online experiments.

Acknowledgment

This research was supported by Zhejiang Provincial Natural Science Foundation of China under No. LY17F020014, the Key Program of Zhejiang Province under No. 2015C01027, and Australian Research Council (ARC) Linkage Project under No. LP140100937.

REFERENCES

[1] A. Oulasvirta, J. P. Hukkinen, and B. Schwartz, "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*. ACM, 2009, pp. 516–523.

[2] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender Systems Handbook*. Springer, 2015, pp. 1–34.

[3] X. Wu, Q. Liu, E. Chen, L. He, J. Lv, C. Cao, and G. Hu, "Personalized next-song recommendation in online karaokes," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 137–140.

[4] D. Wang, S. Deng, and G. Xu, "Gemrec: A graph-based emotion-aware music recommendation approach," in *International Conference on Web Information Systems Engineering*. Springer, 2016, pp. 92–106.

[5] P. Chiliguano and G. Fazekas, "Hybrid music recommender using content-based and social information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 2618–2622.

[6] M. R. Smith, M. S. Gashler, and T. Martinez, "A hybrid latent variable neural network model for item recommendation," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[7] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, "Learning to embed songs and tags for playlist prediction," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, pp. 349–354.

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[9] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*. Springer, 2007, pp. 291–324.

[10] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Springer, 2007, pp. 325–341.

[11] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[12] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latenttopic sequential patterns," in *Proceedings of the Sixth ACM Conference on Recommender Systems*. ACM, 2012, pp. 131–138.

[13] S. Deng, D. Wang, X. Li, and G. Xu, "Exploring user emotion in microblogs for music recommendation," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9284–9293, 2015.

[14] C. Luo, X. Cai, and N. Chowdhury, "Probabilistic temporal bilinear model for temporal dynamic recommender systems," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[15] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008, pp. 263–272.

[16] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 452–461.

[17] T. Zhao, J. Hu, P. He, H. Fan, M. Lyu, and I. King, "Exploiting homophily-based implicit social network to improve recommendation performance," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 2539–2547.

[18] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 701–710.

[19] Y. Pang, Z. Ji, P. Jing, and X. Li, "Ranking graph embedding for learning to rerank," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1292–1303, 2013.

[20] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 1067–1077.

[21] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 891–900.

[22] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015, pp. 2111–2117.

[23] M. G. Carneiro, T. H. Cupertino, and L. Zhao, "K-associated optimal network for graph embedding dimensionality reduction," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 1660–1666.

[24] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester, "Community-based question answering via heterogeneous social network learning," in *AAAI*, 2016, pp. 122–128.

[25] X. Geng, H. Zhang, J. Bian, and T.-S. Chua, "Learning image and user features for recommendation in social networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4274–4282.

[26] X. Zhang, W. Hu, S. Chen, and S. Maybank, "Graph-embedding-based learning for robust object tracking," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1072–1084, 2014.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[28] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 891–900.

[29] D. Wang, S. Deng, S. Liu, and G. Xu, "Improving music recommendation using distributed representation," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 125–126.

[30] L. A. Adamic and B. A. Huberman, "Power-law distribution of the world wide web," *Science*, vol. 287, no. 5461, pp. 2115–2115, 2000.

[31] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun, "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*. ACM, 2010, pp. 723–732.

[32] S. Kabbur, X. Ning, and G. Karypis, "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*. ACM, 2013, pp. 659–667.

[33] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. ACM, 1994, pp. 175–186.

[34] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463.

[35] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[36] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 627–636.

[37] J. Li, H. Xu, X. He, J. Deng, and X. Sun, "Tweet modeling with lstm recurrent neural networks for hashtag recommendation," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1570–1577.