CAME: Content- and Context-Aware Music Embedding for Recommendation

Dongjing Wang, Xin Zhang, Dongjin Yu, *Member, IEEE*, Guandong Xu, *Member, IEEE*, and Shuiguang Deng, *Member, IEEE*

***Abstract*— Traditional recommendation methods suffer from limited performance, which can be addressed by incorporating abundant auxiliary/side information. This article focuses on a personalized music recommender system that incorporates rich content and context data in a unified and adaptive way to address the abovementioned problems. The content information includes music textual content, such as metadata, tags, and lyrics, and the context data incorporate users' behaviors, including music listening records, music playing sequences, and sessions. Specifically, a heterogeneous information network (HIN) is first presented to incorporate different kinds of content and context data. Then, a novel method called content- and context-aware music embedding (CAME) is proposed to obtain the low-dimension dense real-valued feature representations (embeddings) of music pieces from HIN. Especially, one music piece generally highlights different aspects when interacting with various neighbors, and it should have different representations separately. CAME seamlessly combines deep learning techniques, including convolutional neural networks and attention mechanisms, with the embedding model to capture the intrinsic features of music pieces as well as their dynamic relevance and interactions adaptively. Finally, we further infer users' general musical preferences as well as their contextual preferences for music and propose a content- and context-aware music recommendation method. Comprehensive experiments as well as quantitative and qualitative evaluations have been performed on real-world music data sets, and the results show that the proposed recommendation approach outperforms state-of-the-art baselines and is able to handle sparse data effectively.**

***Index Terms*— Attention, content, context-aware, embedding, recommender systems.**

I. Introduction

THE digital music market has a rapid growth, which benefits from innovation on mobile internet technologies as well as smart digital devices. Based on the 2019 International Federation of Phonographic Industry (IFPI) Global Music Report,[1] the global recorded music market has achieved growth by 9.7% in 2018, which is the fourth consecutive year of growth. Specifically, 47% of the global revenue increase is driven by online music revenue. Today, smart mobile devices are capable of storing thousands of music pieces, and mobile applications allow users to access millions of music conveniently via mobile Internet. Meanwhile, it becomes more and more difficult for users to obtain the music pieces that meet their preferences.

Recommender systems [1]–[4] are proposed to decrease the search costs by helping users to find the relevant items from a huge amount of online content. The algorithmic advances of recommendation methods applied in various fields [5], [6] also have improved the performance of music recommendation. However, the traditional approaches usually suffer from problems, such as low accuracy and data sparsity, especially for a huge number of music pieces. Hybrid recommendation approaches [7], [8] are developed to alleviate these problems by combining traditional recommendation methods with supplementary information, such as associated textual descriptions and item metadata. Nevertheless, existing hybrid methods cannot fully exploit interactive/context data and content information in a unified and adaptive way.

Specifically, music listening is typical contextual behavior, and the contexts can help predict users' preferences precisely as well as perform accurate music recommendation. In general, the popularity of smartphones enables people to listen to music almost anywhere at any time, which makes the dynamic contexts difficult to obtain directly. Besides, music content data, such as metadata, description, and lyrics, contain various useful information, which can help learn the feature representation of music and infer users' musical preferences. Furthermore, one music piece generally highlights specific aspects dynamically when listened to together with different music pieces or by different listeners. For instance, a piece of pop–rock music may present more rock and roll features when it is played together with other rock music pieces, and the same music piece may show its pop styles when listened to by pop music fans. Therefore, how to fully exploit rich context information and kinds of content data is a key factor to achieve better recommendations.

Based on the abovementioned analysis, we propose a content- and context-aware music recommendation model that can exploit heterogeneous information to perform precise music recommendation. Specifically, inspired by the

Manuscript received July 30, 2019; revised December 23, 2019 and March 7, 2020; accepted March 28, 2020. Date of publication April 14, 2020; date of current version March 1, 2021. This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ20F020015 and in part by the Fundamental Research Funds for the Provincial University of Zhejiang under Grant GK199900299012-017. *(Corresponding author: Dongjing Wang.)*

Dongjing Wang, Xin Zhang, and Dongjin Yu are with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: dongjing.wang@hdu.edu.cn; zhangxin@hdu.edu.cn; yudj@hdu.edu.cn).

Guandong Xu is with the Advanced Analytics Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: guandong.xu@uts.edu.au).

Shuiguang Deng is with the School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: dengsg@zju.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2984665

[1] https://www.ifpi.org/news/IFPI-GLOBAL-MUSIC-REPORT-2019

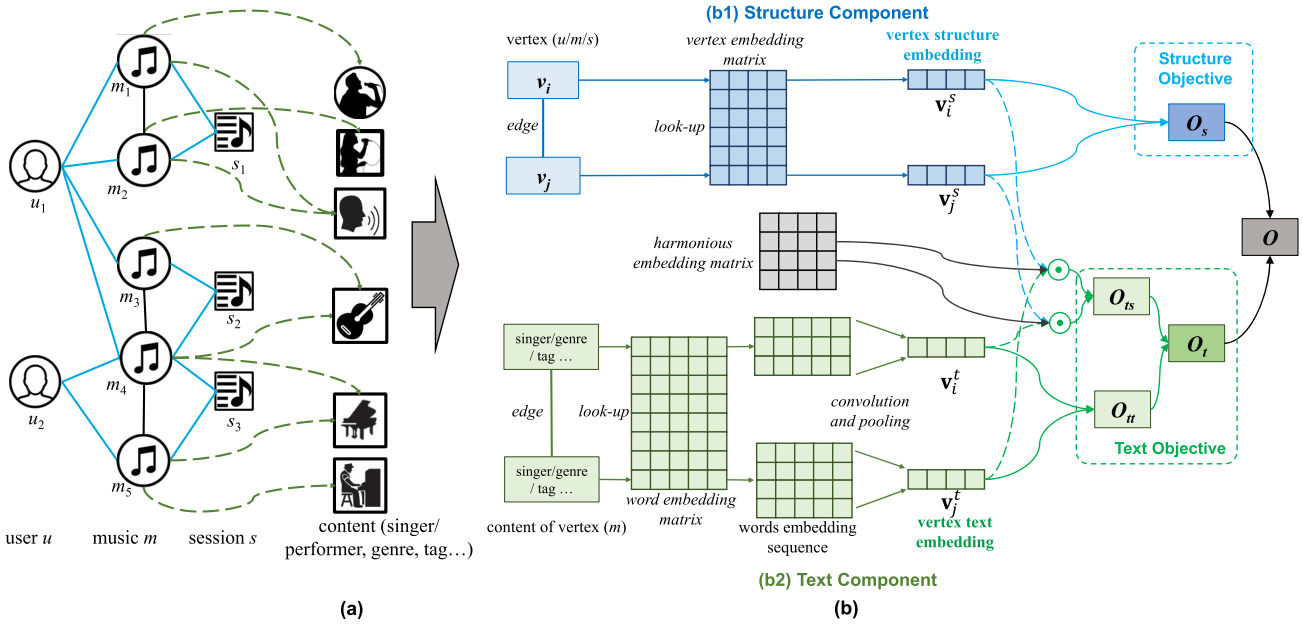

Fig. 1. Overall framework of the proposed CAME approach. HIN can incorporate different kinds of data and information. Then, the structure component in CAME is used to obtain the structure feature representation (structure embedding) from behavior information in HIN, such as users' listening behaviors, music playing sequences, and sessions. Besides, the text component in CAME is able to incorporate content information in HIN, including singer, album, tag, description, and lyrics, as well as learn text embeddings for music vertices. (a) HIN. (b) Model framework and objective.

collaborative filtering (CF) method based on matrix factorization, we propose a model to learn the latent real-value low-dimension feature representations from both interactive/context data and textual content data. The framework of the proposed approach is shown in Fig. 1. First, we incorporate different kinds of data and information into a unified model, namely, heterogeneous information network (HIN). Specifically, content information includes music textual content, such as metadata, tags, and lyrics, and the context data incorporate users' behaviors, including music listening records, music playing sequences, and sessions. Then, we propose a content- and context-aware music embedding (CAME) method to obtain the low-dimension dense real-valued feature vector (embedding) of music pieces from HIN. Especially, we consider that one music piece generally highlights different aspects when played together with various kinds of music, and it should have different representations separately. CAME can learn the content- and context-aware embeddings of music pieces via network embedding and convolutional neural networks (CNNs) with attention mechanism and is able to model the intrinsic features of music pieces as well as their relevance and interactions precisely. Finally, we infer users' general musical preferences as well as their contextual preferences for music and propose a content- and context-aware music recommendation method.

Compared with existing methods, the proposed approach is capable of: 1) incorporating and leveraging interactive context data and textual content information to alleviate the data sparsity problem; 2) adaptively coping with various aspects of items (music) when interacting with different neighbors; and 3) capturing dynamic features and relevance from heterogeneous information precisely to further improve the performance of recommendation.

We summarize the main contributions of this article as follows.

1) We present HIN that can encode interactive/context data and textual content data in a unified and flexible way.
2) We propose a novel CAME model that seamlessly combines deep learning techniques, including CNN and attention mechanism, with embedding model to learn the content- and context-aware low-dimensional representations (embeddings) of music pieces from HIN precisely and adaptively for a better recommendation.
3) Comprehensive experiments, including quantitative and qualitative evaluations, are conducted on real-world data sets, which demonstrate that the proposed model outperforms state-of-the-art baselines and is able to handle sparse data effectively.

The rest of this article is organized as follows. In Section II, we discuss the related works. Problem definitions and the proposed approach are presented in detail in Sections III and IV. The experimental results and analysis are described in Section V. In Section VI, we conclude a summary and give future work.

II. Related Work

We introduce related works from three aspects, including music recommendation, network embeddings, and the attention mechanism that inspire this work.

A. Music Recommendation

Generally, existing works on music recommendation can be divided into CF methods, content-based recommendation methods, context-aware methods, and hybrid recommendation methods [9].

Specifically, CF methods [10] can be categorized into user-based CF (UCF) and item-based CF (ICF). UCF approaches estimate the relevance between users based on users' behaviors and perform music recommendations according to similar users' musical preferences, while ICF methods perform recommendations based on similarity or relevance between items. Content-based recommendation approaches [11], [12] conduct recommendations mainly based on music content information as well as the profiles of users' preferences, which can be inferred from users' historical listening records. Context-aware recommendation methods [13], [14] incorporate the contexts related to environments or users to achieve better recommendation. The environment-related contexts include temporal contexts [15], geographical contexts [16], and so on. The user-related contexts include activity [17], emotional state [14], and so on. Hybrid methods [8] try to alleviate the influence of data sparsity by integrating different recommendation strategies and usually obtain better results.

Lately, more and more works try to utilize users' implicit feedbacks in CF approaches [18], [19]. Compared with explicit feedbacks, implicit feedbacks are usually more abundant and easier to obtain. Besides, there are some works that try applying long short-term memory (LSTM) neural networks [20] and knowledge graph [21] in music recommendation, and experiments confirm their effectiveness.

In this article, we focus on leveraging content and context information by an HIN and use deep learning techniques to capture dynamic features and relevance between music pieces adaptively to further improve the performance of recommendation.

B. Network Embedding

This work is also inspired by approaches of network embedding [22]–[25] in general, which can map symbolic data from a high-dimensional space with the dimension equal to the number of data objects to a low-dimensional real-valued vector space. By effectively capturing items' intrinsic features and relationships in the training set, the learned embeddings can alleviate dimensional disaster and the problem of data sparsity.

Perozzi *et al.* [23] present DeepWalk that adopts a revised random walk algorithm to learn the representation of vertices in the information network. In this method, vertices with similar structures have similar embeddings. Based on DeepWalk, Yang *et al.* [25] propose an improved model named text-associated DeepWalk (TADW) that takes the textual attributes into consideration when learning the representation of vertices. Tang *et al.* [24] propose a large-scale information network embedding (LINE) method that combines the first- and second-order information to obtain the representation of vertices efficiently. The abovementioned methods have achieved effective results on various tasks.

Recently, the techniques of network embedding are used by lots of tasks, such as dimensionality reduction [26], knowledge transfer [27], relation modeling [28], recommendation [29], visual word mergence [30], object tracking [31], and question answering [32].

C. Attention Mechanism

The attention mechanism in deep learning draws on the way of human's attention, and it has shown its effectiveness in lots of fields and applications, such as computer vision [33], natural language processing [34], [35], and prediction [36], [37]. Specifically, it is a weighted sum strategy that can automatically and adaptively analyze which part of data or information in the input is more significant. For instance, the attention mechanism in machine translation is able to help measure the relevance between words in source sentences and words in target sentences. In brief, attention makes neural networks more explainable and adaptive.

Recently, more works also try applying attention mechanisms in recommender systems. For example, Pei *et al.* [38] propose a model named interacting attention-gated recurrent network (IARN) that uses attention mechanism to discriminate the correlation between users and items, which increases the interpretability of recommendation results. Miura *et al.* [39] combine various kinds of information, including textual content and user networks using attention model and recurrent neural network (RNN) in geolocation prediction, which outperforms previous ensemble approaches. Chen *et al.* [40] combine implicit feedback and a CF framework together with attention model in both item level and component level for an accurate multimedia recommendation. Attentional factorization machines (AFMs) [41] adopt attention models to measure the significance and relevance between different features as well as their interactions in factorization machines (FMs). Wang *et al.* [42] present an attentive deep model for better article recommendation, which uses multiple text models to adaptively capture important features of each article and then utilizes attention-based network architecture to dynamically assign influence factors on different models and effectively learn editors' dynamic selection criteria. Han *et al.* [43] propose a deep neural network-based recommendation framework to learn the adaptive representations of users, and experiment results show its effectiveness in the accurate recommendation.

III. Definition

The definitions of key notations and symbols used in this article are given in Table I. Formally, we define user set as $U = \{u_1,\ u_2,\ u_3 \ldots,\ u_{|U|}\}$ and music set as $M = \{m_1,\ m_2,\ m_3 \ldots,\ m_{|M|}\}$, where $|U|$ and $|M|$ denote the number of unique users and music pieces, respectively. $H^u = \{m^u_1,\ m^u_2,\ m^u_3 \ldots,\ m^u_{|H^u|}\}$ represents user u's historical music playing sequence, and each music record $m^u_i \in M$ in the sequence has corresponding time and devices information.

Furthermore, user u's listening history H^u can be divided into different sessions $S^u = \{S^u_1,\ S^u_2,\ S^u_3 \ldots,\ S^u_{|S^u|}\}$ according to time and device information. Specifically, user u's nth session is defined as $S^u_n = \{m^u_{n,1},\ m^u_{n,2} \ldots,\ m^u_{n,|S^u_n|}\}$, where $m^u_{n,j} \in M$. An example is given in Fig. 2. H^u consists of eight music pieces that are ordered according to their timestamps. Obviously, m^1_u, m^2_u, and m^3_u can be aggregated into session S^u_1, and the other five music pieces can be assigned to session S^u_2. Formally, $S^u = \{S^u_1,\ S^u_2\}$ is user u's session set, where $S^u_1 = \{m^u_1, m^u_2, m^u_3\}$ and $S^u_2 = \{m^u_4, m^u_5, m^u_6, m^u_7, m^u_8\}$. Note

TABLE I
SYMBOLS USED IN THIS ARTICLE

Notation	Interpretation
$N = (V, E, W)$	the heterogeneous information network
V	vertex xet
E	edge set
W	edge weight set
$U \subseteq V$, $M \subseteq V$	user set and music set
H	all users' historical listening sequences
H^u	user u's historical music listening sequence
$S^u \subseteq V$	user u's session set
S_i^u	the i-th session of user u
$C \subseteq V$	the set of music content (singer, album, tag, description, lyrics, and so on)
$\mathbf{v}$	the embedding
$\mathbf{p}_g^u$, $\mathbf{p}_c^u$	user u's general musical preferences and contextual preferences for music

that Fig. 2 omits some information, such as playing devices and description of music for simplicity.

Therefore, the recommendation task is defined as how to recommend appropriate music piece to target user u according to her/his historical records H^u and S^u and content information of music.

IV. METHODOLOGY

The method that we present is composed of three components: 1) HIN for incorporating various information; 2) CAME for feature representation learning; and 3) content- and context-aware music recommendation approach.

A. Heterogeneous Information Network

In order to learn the content- and context-aware embeddings of music pieces from kinds of information, an HIN is presented to incorporate the relationships between music pieces and users as well as the content of music in a unified manner. The definition of HIN and its edges are given as follows.

Definition 1: The user–music edges $E_{u,m} \subseteq E$ are links between users and the music pieces they have listened to, and E is the edge set in HIN. User–music edges encode correlations between music at the user level, which is similar to the idea in CF approaches. For example, two music pieces may be quite similar to each other if they are listened to by common users, and the user–music edges in HIN can represent such kind of information effectively. Besides, the user–music edges also indicate users' specific musical preferences as well as the intrinsic features of music pieces. The user–music weight $w_{u,m} \in [0, 1)$ is defined with hyperbolic tangent sigmoid function as $w_{u,m} = \tanh(w') = (e^{w'} - e^{-w'})/(e^{w'} + e^{-w'})$, where $w' \in (0, \infty)$ is the frequency how often user u has listened to music m.

Definition 2: The session–music edges $E_{s,m} \subseteq E$ are links between sessions and the music pieces that appear in the corresponding sessions. Generally, each user has specific preferences for some time (during the session), and similar music pieces are more likely to appear in the same sessions. In other words, $E_{s,m}$ encodes music co-occurrence information at the session level, which is equivalent to the idea in context-/session-aware recommendation approaches. Similarly, the weight of the session–music edge $w_{s,m} \in [0, 1)$ is defined as $w_{s,m} = \tanh(w') = (e^{w'} - e^{-w'})/(e^{w'} + e^{-w'})$, where $w' \in (0, \infty)$ is the frequency that music piece m appears in session s.

Definition 3: The music–music edges $E_{m,m} \subseteq E$ are links between music pieces that are listened to together. Since users' preferences are relatively fixed especially during a short period of time, music pieces that are close in the music listening sequences generally have common styles or features. In other words, music co-occurrence information indicates music pieces' features as well as users' preferences. The music–music edge weight $w_{m,m} \in [0, 1)$ is defined as $w_{m,m} = \tanh(w') = (e^{w'} - e^{-w'})/(e^{w'} + e^{-w'})$, where $w' \in (0, \infty)$ is the frequency how often music m_i and music m_j appear in the same context window of music sequence. In Fig. 2, the size of context window is set as 2.

Definition 4: The music–content edges $E_{m,c} \subseteq E$ indicate links between music pieces and their content features, including singers, albums, tags, text description, and lyrics. The music–content edge weight between music piece m and its corresponding content c_m is set to be 1.

Definition 5: The HIN is defined as $N = (V, E, W)$, where $V = (U, S, M, C)$ represents the vertex set, and U, S, M, and C are the user set, session set, music set, and content set, separately. E is the set of edges, including user–music edges $E_{u,m}$, session–music edges $E_{s,m}$, music–music edges $E_{m,m}$, and music–content edges $E_{m,c}$, which are defined earlier. W is the set of corresponding edges' weights. Especially, the music–content edges are static since the contents of music remain basically unchanged, while the user–music edges, session–music edges, and music–music edges will increase dynamically as more interaction data are collected. Besides, the HIN can be extended flexibly to incorporate various kinds of data, such as social networks or music playlist.

B. Content- and Context-Aware Music Embedding

Given the context (user–music edge, session–music edge, and music–music edge) and content features (music–content edge), the CAME model is proposed to learn the feature vectors (embeddings) of music pieces from HIN. Specifically, the context information and content data in HIN indicate music pieces' features and their correlations, and the proposed CAME can effectively make use of this information. Especially, we consider that one music piece generally highlights different aspects when played together with various kinds of music, and it should have different representations. The proposed approach CAME can learn the content- and context-aware embeddings of music pieces precisely via CNNs with attention, which enables CAME to model the dynamic relevance between music pieces as well as their interactions adaptively.

Specifically, we adopt two types of embeddings for a vertex $v \in V$ in HIN, i.e., structure embedding $\mathbf{v}^s$ and text embedding $\mathbf{v}^t$. Specifically, structure embeddings can capture the context

Music Playing Data of user u and Metadata

No.	H^u	S^u	Music	Playing Time	Singer	Album	Tag	…
1	m_u^1	S_1^u	My Heart Will Go On	20150921 16:51	Celine Dion	Let's Talk About Love	pop, female vocalists, love, soundtrack	…
2	m_u^2	S_1^u	Immortality	20150921 16:55	Celine Dion	My Love Essential Collection	pop, female vocalists, ballad	…
3	m_u^3	S_1^u	Hero	20150921 16:59	Mariah Carey	#1's	pop, female vocalists, 90s, ballad	…
4	m_u^4	S_2^u	I Do It for You	20150923 19:56	Bryan Adams	So Far So Good	rock, ballad, classic rock, pop	…
5	m_u^5	S_2^u	Thank You for Loving Me	20150923 20:03	Bon Jovi	Crush	rock, classic rock, ballad, hard rock	…
6	m_u^6	S_2^u	It's My Life	20150923 20:08	Bon Jovi	Crush	rock, classic rock, hard rock	…
7	m_u^7	S_2^u	Enter Sandman	20150924 10:49	Metallica	The Metallica Collection	rock, metal, heavy metal, thrash metal	…
8	m_u^8	S_2^u	Nothing Else Matters	20150924 10:54	Metallica	The Metallica Collection	rock, metal, heavy metal, ballad	…

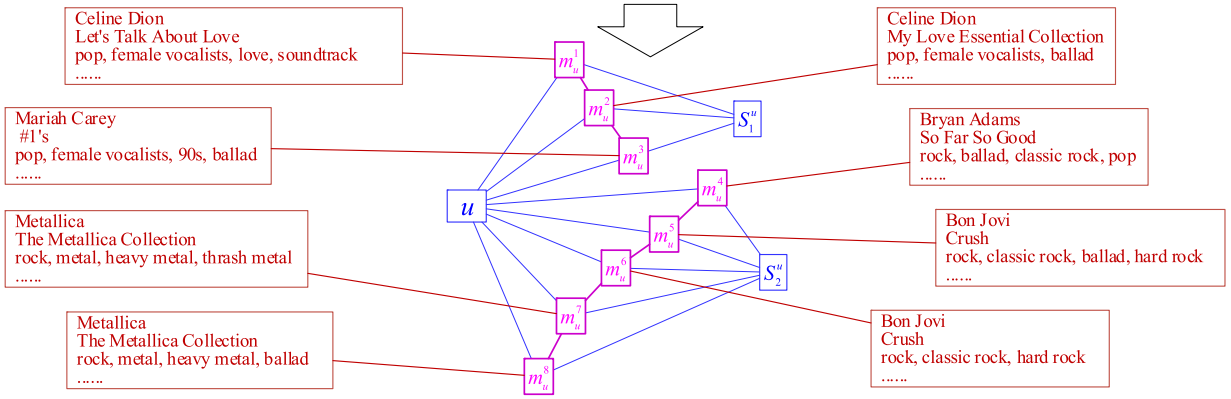

Fig. 2. Example of creating an HIN from interactive/context and textual content data. $H^u = \{m_1^u, m_2^u, \ldots, m_8^u\}$ is user u's historical music playing sequence, and u's listening history H^u is aggregated into two sessions S_1^u and S_2^u.

structure information in the HIN, while text embeddings can capture the textual meanings lying in vertex v's associated content information.

As shown in Fig. 1, the objective function of CAME consists of structure objective and text objective. The structure objective incorporates edge information between user/music/session vertices in HIN, and vertices with common neighbors/edges will yield similar structure embeddings. The text objective models the relationship between music pieces and their content information using CNNs with attention dynamically. Especially, the attention mechanism enables CAME to adaptively cope with various aspects of music when interacting with different neighbors. Generally, music vertices with common textual contents will have similar text embeddings.

Formally, CAME aims to maximize the overall objective of edges, and the objective function is defined as

$$O = \sum_{e \in E} O_s(e) + \sum_{e \in E_{m,m}} O_t(e) + \lambda \|\Theta\|_2^2 \tag{1}$$

where $O_s(e)$ is the structure objective and $O_t(e)$ represents the text objective. Specifically, only music pieces have contents, so the text objective is calculated only on music–music edges $E_{m,m} \subseteq E$. $\|\Theta\|_2^2$ is the regularization term of all parameters Θ, and λ is the weight of regularization term. In the following, we will introduce the structure objective and the text objective in detail.

1) Structure Objective: The structure objective aims to measure the log-likelihood of an edge using the structure embeddings. For an edge e between vertices $v_i, v_j \in V$ in HIN, we define the corresponding joint probability with softmax function as follows:

$$p^s(v_i, v_j) = \frac{\exp\left(\mathbf{v}_i^{s\top} \cdot \mathbf{v}_j^s\right)}{\sum_{v_k \in V} \exp\left(\mathbf{v}_k^{s\top} \cdot \mathbf{v}_j^s\right)} \tag{2}$$

where $\mathbf{v}_i^s \in \mathcal{R}^d$ and $\mathbf{v}_j^s \in \mathcal{R}^d$ are the structure embeddings of vertices v_i and v_j, and d is the dimension of structure embeddings. $p^s(\cdot,\cdot)$ in (2) is the distribution over vertex space $V \times V$. Then, we minimize the Kullback–Leibler (KL) divergence between two distributions $p^s(\cdot,\cdot)$ and $\hat{p}^s(\cdot,\cdot)$ to retain the structure and context information in HIN, and the corresponding cost function is defined as

$$O_s(e) = -d_{\mathrm{KL}}(\hat{p}^s(\cdot,\cdot), p^s(\cdot,\cdot)) \tag{3}$$

where $\hat{p}^s(\cdot,\cdot)$ is the empirical distribution. Specifically, the empirical distribution between two vertices $v_i \in V$ and $v_j \in V$ should encode the structure and context information in HIN, which is formally defined as

$$\hat{p}^s(v_i, v_j) = \frac{w_{i,j}}{\sum_{v_k \in V} w_{k,j}} \tag{4}$$

where $w_{i,j} \in W$ denotes the edge weight between vertices v_i and v_j in HIN.

Based on (3) and (4), the structure objective function can be defined as

$$\begin{aligned} O_s(e) &= -d_{\mathrm{KL}}(\hat{p}^s(\cdot,\cdot), p^s(\cdot,\cdot)) \\ &= -d_{\mathrm{KL}}(\hat{p}^s(v_i, v_j), p^s(v_i, v_j)) \\ &= -\hat{p}^s(v_i, v_j) \log \frac{\hat{p}^s(v_i, v_j)}{p^s(v_i, v_j)} \\ &\propto w_{i,j} \log p^s(v_i, v_j). \end{aligned} \tag{5}$$

Then, the structure embeddings can be learned via maximizing the structure objective function defined earlier.

2) Text Objective: The text objective is used to incorporate content information, including singer, album, tag, description, lyrics, and so on, and learn text embeddings for vertices. The text objective consists of two parts, which is formally defined as

$$O_t(e) = O_{tt}(e) + O_{ts}(e) \tag{6}$$

where O_{tt} measures the log-likelihood of a music–music edge using the text embeddings learned from music content and O_{ts} further maps the text embeddings and structure embeddings into the same representation space. Formally, O_{tt} is defined with KL divergence as

$$\begin{aligned} O_{tt}(e) &= -d_{\mathrm{KL}}(\hat{p}^t(\cdot,\cdot), p^t(\cdot,\cdot)) \\ &\propto w_{i,j} \log p^t(v_i, v_j) \end{aligned} \tag{7}$$

where $p^t(\cdot,\cdot)$ and $\hat{p}^t(\cdot,\cdot)$ are joint distribution and empirical distribution, and $w_{i,j}$ represents the weight of edge between two music vertices $v_i \in M$ and $v_j \in M$ in HIN. Formally, $p^t(v_i, v_j)$ is defined as

$$p^t(v_i, v_j) = \frac{\exp\left(\mathbf{v}_i^{t\top} \cdot \mathbf{v}_j^t\right)}{\sum_{v_k \in M} \exp\left(\mathbf{v}_k^{t\top} \cdot \mathbf{v}_j^t\right)} \tag{8}$$

where $\mathbf{v}_i^t \in \mathcal{R}^d$ and $\mathbf{v}_j^t \in \mathcal{R}^d$ are text embeddings for v_i and v_j, respectively. O_{ts} tries mapping text embeddings and structure embeddings into the same space but does not constrain them to be identical for the consideration of their own characteristics.

Similarly, O_{ts} is formally defined with KL divergence as

$$\begin{aligned} O_{ts}(e) &= -d_{\mathrm{KL}}(\hat{p}^{ts}(\cdot,\cdot), p^{ts}(\cdot,\cdot)) \\ &\propto w_{i,j} \log p^{ts}(v_i, v_j) \end{aligned} \tag{9}$$

where $p^{ts}(\cdot,\cdot)$ and $\hat{p}^{ts}(\cdot,\cdot)$ are joint distribution and empirical distribution based on content and context information, and $p^{ts}(v_i, v_j)$ is defined as

$$p^{ts}(v_i, v_j) = \frac{\exp\left(\mathbf{v}_i^{t\top} \cdot \mathbf{H}^{ts} \cdot \mathbf{v}_j^s\right)}{\sum_{v_k \in M} \exp\left(\mathbf{v}_k^{t\top} \cdot \mathbf{v}_j^s\right)} + \frac{\exp\left(\mathbf{v}_i^{s\top} \cdot \mathbf{H}^{st} \cdot \mathbf{v}_j^t\right)}{\sum_{v_k \in M} \exp\left(\mathbf{v}_k^{s\top} \cdot \mathbf{v}_j^t\right)} \tag{10}$$

where $\mathbf{v}^t$ and $\mathbf{v}^s$ are text embedding and structure embedding of music vertex $v \in M$, and $\mathbf{H}^{ts} \in \mathcal{R}^{d\times d}$ and $\mathbf{H}^{st} \in \mathcal{R}^{d\times d}$ are the harmonious embedding matrices that help coordinate structure and text space. Specifically, $\mathbf{H}^{ts} \in \mathcal{R}^{d\times d}$ helps harmonize the text embedding to structure latent space, and $\mathbf{H}^{st} \in \mathcal{R}^{d\times d}$ helps harmonize the structure embedding to text latent space.

The structure embeddings are trainable parameters, and text embeddings are learned from associated textual content information (words and sentences) of vertices. In this work, we combine CNNs [44], [45] with attention mechanism [46] to capture the local semantic dependence among words and emphasize those words that indicate music pieces' intrinsic features as well as their relevance and then obtain the text embeddings effectively. Specifically, the contents of all music are preprocessed based on words' term frequency–inverse document frequency (TF-IDF) [47], and the words with lower TF-IDF value will be removed from the data set. As shown in Fig. 3, the CNN obtains the text embeddings through three layers: looking-up layer, convolution layer, and pooling layer.

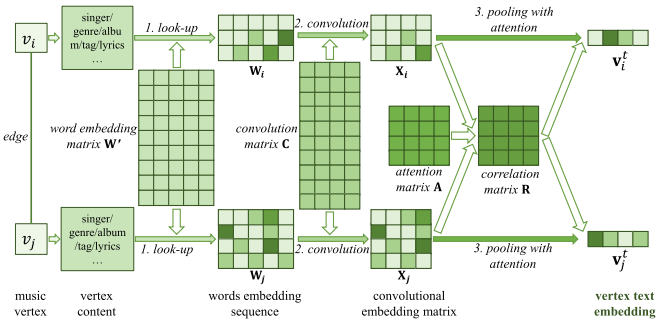

Fig. 3. Illustration of learning text embeddings using CNNs with attention mechanism.

The looking-up layer transforms each word w_i in the content sentences of music piece into corresponding word embeddings $\mathbf{w} \in \mathcal{R}^{d'}$ with a word embedding matrix $\mathbf{W}' \in \mathcal{R}^{d'\times m'}$, where d' denotes the dimension of word embeddings and m' is the size of the whole word vocabulary. Then, each music piece's content sequence with m words is represented with words embedding sequence $\mathbf{W} = (\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_m)$.

The convolution layer extracts local features from words embedding sequence $\mathbf{W} \in \mathcal{R}^{d'\times m}$ by performing convolution operation with a convolution matrix $\mathbf{C} \in \mathcal{R}^{d\times(l\times d')}$, and l denotes the size of the sliding convolution window. Specifically, zero padding vectors are added at the edge of the content sentence, so the output of the convolution layer for content sequence of each music piece is a convolutional embedding matrix $\mathbf{X} \in \mathcal{R}^{d\times m}$.

In the pooling layer, the mean-pooling strategy together with attention mechanism is applied over the embedding matrix to obtain the text embedding $\mathbf{v}_i^t$ of each music vertex $v_i \in M$. Specifically, one music piece generally highlights different aspects when played together with various kinds of music, and it should have different representations separately. The attention mechanism enables the pooling layer to be aware of music vertices pairs (edges) as well as their dynamic features in the HIN and learns the specific representation of music pieces adaptively. For example, the text embedding of a piece of pop–rock music may show more rock and roll features when it is played together with rock music pieces, and the text embedding of the same music piece may show its pop features when listened to together with pop music pieces. In the following, we will illustrate how to learn the text embedding with music–music edge $e \in E_{m,m}$ and its music vertices $v_i \in M$ and $v_j \in M$.

First, we obtain the words embedding sequence $\mathbf{W}_i = (\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_m)$ and $\mathbf{W}_j = (\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_n)$ via the looking-up layer, where $\mathbf{W}_i \in \mathcal{R}^{d'\times m}$, $\mathbf{W}_j \in \mathcal{R}^{d'\times n}$, m and n are the length of corresponding content sequence, and d' is the dimension of word embedding.

Second, the convolution layer (with zero padding vectors) extracts local features from word embedding matrix $\mathbf{W}_i \in \mathcal{R}^{d'\times m}$ and $\mathbf{W}_j \in \mathcal{R}^{d'\times n}$, and d' and get convolutional embedding matrix $\mathbf{X}_i \in \mathcal{R}^{d\times m}$ and $\mathbf{X}_j \in \mathcal{R}^{d\times n}$.

Third, an attentive matrix $\mathbf{A} \in \mathcal{R}^{d \times d}$ is introduced to compute the correlation matrix between convolutional embedding matrix $\mathbf{X}_i$ and $\mathbf{X}_j$, which is defined as

$$\mathbf{R} = \tanh(\mathbf{X}_i{}^{\top} \cdot \mathbf{A} \cdot \mathbf{X}_j) \tag{11}$$

where $\mathbf{R}[i, j]$ represents the correlation between music vertices' corresponding words. Especially, the attentive matrix $\mathbf{A}$ enables the correlation matrix $\mathbf{R}$ to capture the dynamic feature correlations between music pieces in HIN.

Fourth, we apply row-pooling operation over $\mathbf{R}$ to generate m-dimension attention vector of vertex v_i as

$$\mathbf{a}_i = \begin{bmatrix} \text{mean}(\mathbf{R}[1,1], \mathbf{R}[1,2], \ldots, \mathbf{R}[1,n]) \\ \text{mean}(\mathbf{R}[2,1], \mathbf{R}[2,2], \ldots, \mathbf{R}[2,n]) \\ \vdots \\ \text{mean}(\mathbf{R}[m,1], \mathbf{R}[m,2], \ldots, \mathbf{R}[m,n]) \end{bmatrix}. \tag{12}$$

Similarly, the n-dimension attention vector of vertex v_j is computed with column-pooling operation as

$$\mathbf{b}_j = \begin{bmatrix} \text{mean}(\mathbf{R}[1,1], \mathbf{R}[2,1], \ldots, \mathbf{R}[m,1]) \\ \text{mean}(\mathbf{R}[1,2], \mathbf{R}[2,2], \ldots, \mathbf{R}[m,2]) \\ \vdots \\ \text{mean}(\mathbf{R}[1,n], \mathbf{R}[2,n], \ldots, \mathbf{R}[m,n]) \end{bmatrix}. \tag{13}$$

Then, the text embedding for music vertices $v_i \in M$ and $v_j \in M$ can be obtained by multiplying the convolutional embedding matrix with attention vectors, which are defined as follows:

$$\begin{aligned} \mathbf{v}_i^t &= \mathbf{X}_i \cdot \mathbf{a}_i \\ \mathbf{v}_j^t &= \mathbf{X}_j \cdot \mathbf{b}_j. \end{aligned} \tag{14}$$

Finally, the content- and context-aware embedding of vertex v in HIN is learned as $\mathbf{v} = \mathbf{v}^t \oplus \mathbf{v}^s$, where $\mathbf{v}^t$ and $\mathbf{v}^s$ are the corresponding text embedding and structure embedding and $\oplus$ is concatenation operation.

3) Learning: In the training stage, CAME minimizes the log probability defined in (2), (8), and (10) over all data in HIN. However, it is impractical to directly optimize the above functions because the computation complexity of the full softmax functions in the abovementioned equations is proportional to the vertex set size $|V|$ (or music vertex set size $|M|$). Therefore, we use negative sampling technique [48] to compute the objective functions approximately and effectively. Specifically, for each edge (v_i, v_j) and the embedding $\mathbf{v}_i$ and $\mathbf{v}_j$ (including structure embedding and text embedding) of corresponding vertices, the negative sampling method computes the original objective approximately with the following objective function:

$$\log \sigma\left(\mathbf{v}_i^{\top} \cdot \mathbf{v}_j\right) + k \cdot E_{i' \sim P_I}\left[\log \sigma\left(-\mathbf{v}_{i'}^{\top} \cdot \mathbf{v}_j\right)\right] \tag{15}$$

where $\sigma(x) = 1/(1 + e^{-x})$ and k represents the number of negative samples, which is a small constant. i' is the negative sample that is drawn according to the noise distribution P_I over all vertices (or music vertices) in HIN modeled by empirical unigram distribution. Then, it becomes feasible to optimize (2), (8), and (10) with the stochastic gradient algorithm. In each step, an existing edge is sampled according to its weight in HIN, and multiple negative edges are sampled from a noise distribution P_I at the same time. Here, we adopt the edge sampling approach and the alias table method used in [24] and [49], and it takes only $O(1)$ time to repeatedly draw edge samples from the same discrete distribution. Moreover, each step of optimization with negative sampling takes $O(d \times t \times (k+1))$ time, where d is the time taking for one sampling, t is the time consumed by each KL-divergence calculation or convolution, and k is negative sample number. Furthermore, the total steps that the whole optimization takes are proportional to the number of edges $|E|$, so the overall time complexity of optimization is $O(d \times t \times (k+1) \times |E|)$. Since d, t, and k depends on parameters that are set as constants when training/testing, the final time complexity is linear to $|E|$.

C. Content- and Context-Aware Music Recommendation

With the learned embeddings, users' general and contextual preferences can be inferred from their historical music listening records [13], [50]. Here, we use two typical aggregation operations, including average pooling and max pooling to obtain users' preferences. The average-pooling strategy assumes that different input embeddings are independent of each other and also keeps the wholeness and smoothness of the inputs with the linear transformation. Max-pooling strategy tries modeling the interactions among the input embeddings and only extracts significant features with nonlinear operations. As for the recommendation, mean pooling focuses on overall interests, and max pooling puts more importance on specific interests.

Specifically, the user u's historical music playing sequence H^u indicates her/his general preferences for music, and it is feasible to infer user's interests via mean-pooling and max-pooling operations over the embeddings of music pieces in H^u. Formally, user u's general preference is defined as

$$\mathbf{p}_g^u = f_{\text{mean}}(H^u) + f_{\max}(H^u) \tag{16}$$

where $f_{\text{mean}}(H^u) = 1/(|H^u|) \sum_{v_i \in H^u} \mathbf{v}_i$, $f_{\max}(H^u) = \{\max(\vec{v}_1[1], \ldots, \vec{v}_{|H^u|}[1]), \ldots, \max(\vec{v}_1[d], \ldots, \vec{v}_{|H^u|}[d])\}$, and d is the dimension of embeddings.

Similarly, the user u's recent music playing sequence S^u indicates her/his contextual preferences for music, and it is feasible to infer user's contextual interests via mean-pooling and max-pooling operations over the embeddings of music pieces in S^u. Formally, user u's contextual preference is defined as

$$\mathbf{p}_c^u = f_{\text{mean}}(S^u) + f_{\max}(S^u). \tag{17}$$

Given user u's general preference $\mathbf{p}_g^u$ and contextual preference $\mathbf{p}_c^u$, u's preferences for each music piece m_i consist of two parts: general preferences and contextual preferences. Formally, u's preferences for m_i are defined as follows:

$$p\left(m_i | \mathbf{p}_g^u, \mathbf{p}_c^u\right) = p\left(m_i | \mathbf{p}_g^u\right) + p\left(m_i | \mathbf{p}_c^u\right) \tag{18}$$

where $p(m_i|\mathbf{p}_g^u) = \cos(\mathbf{v}_{m_i}, \mathbf{p}_g^u)$ and $p(m_i|\mathbf{p}_c^u) = \cos(\mathbf{v}_{m_i}, \mathbf{p}_c^u)$ are u's general and contextual preferences for music m_i, respectively, and $\mathbf{v}_{m_i}$ is the embedding of music m_i. Specifically, the embeddings (vectors) learned by the proposed model can effectively represent music pieces' features and users' general/contextual preferences, and cosine metric $\cos(\cdot, \cdot)$ is

TABLE II
STATISTICAL INFORMATION OF THE DATA SET

#(User)	#(Music)	#(Listening)	#(Singer)	#(Album)	#(Tag)
4,284	361,861	4,284,000	38,128	115,219	54,689

used for similarity measure between embeddings (vectors) because of its effectiveness and efficiency.

Finally, we can rank music pieces according to (18) and recommend the top n music pieces to the user.

V. EXPERIMENTS

We conduct comprehensive experiments as well as quantitative and qualitative evaluations to show the effectiveness of the proposed approach CAME. First, we visualize the learned embeddings (including structure embedding and text embedding) in 2-D space. Then, we investigate the relevance between the embeddings' dimension and the recommendation performance and also compare CAME against baselines. Finally, we evaluate how the sparsity of data set influences recommendation accuracy.

A. Experimental Designs

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

1) Data Set: We use a real-world data set crawled from Xiami Music,[2] which provides online music streaming. As illustrated in Table II, the data set after preprocessing is composed of 4 284 000 interactions between 4284 users and 361 861 music pieces as well as the content information, such as singer, album, tag, description, language, and lyrics. Specifically, the contents of all music are preprocessed based on words' TF-IDF [47], and the words with lower TF-IDF value will be removed from the data set. Besides, the average length of the content sequence for each music is 32.9, and the average number of music pieces for each user and each session is 1000 and 25.9, respectively.

Furthermore, Fig. 4 illustrates popularity information (logarithm) of music pieces as well as the relationship between the frequency and quality of textual contents (artist, album, tag, and so on). The results are consistent with the power-law distribution [51], and textual content data can be utilized to help learn the effective embedding as well as improve the performance of recommendation.

The whole data set is divided into two nonoverlapping sets, i.e., the training set and testing set. The testing set contains only 20% of users' second half historical music playing sequences, while all the remaining data are divided into a training set. Note that the evaluation is performed based on fivefold cross-validation.

2) Baselines: We compare the proposed approach with nine state-of-the-art baselines, and the comparison of the features and recommendation strategy between each method is shown in Table III. Specifically, traditional methods, such as UserKNN, BPR, and FISM, only utilize the interactions between users and items to carry out CF recommendations. IPF combines temporal with traditional methods to implement context-aware recommendations. FPMC, HRM, and CSM-UK can extract correlations between items and behaviors as well as the implicit contextual information from sequence information. MEM, HIGE, and CAME can incorporate heterogeneous information into a recommendation, users' music listening records, music playing sequences, sessions, and music textual content, which can help further improve the performance and alleviate data sparsity problem. Particularly, the user–music edges in HIN enable CAME to model interactions between users and music and incorporate the idea of CF. Besides, the proposed approach CAME can fully exploit various textual content information with CNN techniques and model music pieces' intrinsic features as well as their dynamic aspects from via attention mechanism precisely and adaptively.

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

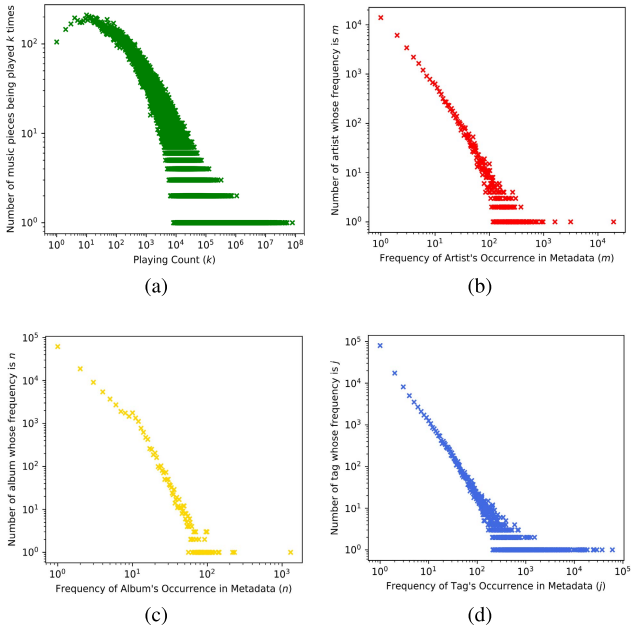

Fig. 4. Popularity analysis of the data set. Popularity analysis of (a) music pieces, (b) artists, (c) albums, and (d) tags.

3) Evaluation Metrics: In recommendation stage, each approach generates a recommendation list of n music pieces ($n = [5, 10, 15, 20, 25, 30]$), denoted by R, which is evaluated by three quality measures, including precision, recall, and F1 score.

Precision is the proportion of relevant music pieces among the recommended music pieces in R, which is defined as

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

where R_i is the recommendation list in the ith recommendation, T_i is the music list that users have listened to, and #(recs) is the total number of recommendations.

Recall is the proportion of relevant music pieces that have been recommended over the total number of relevant music

TABLE III
COMPARISON BETWEEN THE PROPOSED APPROACH CAME AND BASELINES

Methods	Feature			Recommendation Model		
	interaction	listening sequence	content feature	content-based	context-aware	collaborative filtering
UserKNN [52]	√	×	×	×	×	√
BPR [18]	√	×	×	×	×	√
FISM [53]	√	×	×	×	×	√
IPF [15]	√	×	×	×	√	√
FPMC [54]	√	√	×	×	√	√
HRM [55]	√	√	×	×	√	√
CSM-UK [56]	√	√	×	×	√	√
MEM [57]	√	√	√	√	√	√
HIGE [58]	√	√	√	√	√	√
CAME (Proposed)	√	√	√	√	√	√

TABLE IV
PARAMETER SETTINGS FOR TRAINING CAME

Parameter	Value	Description
Context Window Size c	3	the size of sliding context window used to generate music-music edges
Dimension d	[50, 300]	the dimension of the learned embeddings
λ	3e-3	the weight of the parameter regularization term
Negative Sample Number	3	the number of "negative items" should be drawn (in order to increase the efficiency of the training progress)
Negative Sample Power	0.75	items with high frequency f are down sampled, the corresponding probability is proportional to $f^{3/4}$
Learning Rate	1e-3	the step size of optimizing the objective function
Epochs	100	the count of training iterations

pieces, which is defined as

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

F1 score (also known as F-score or F-measure) considers both precision and recall to evaluate the performance of recommendation, which is defined as

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

4) Parameter Settings and Experiment Environment: The detailed configurations of important parameters in CAME and the corresponding descriptions are given in Table IV.

Specifically, the context window size c plays an important role in producing high-quality music embedding, and the experimental results of c with embedding dimension as 50 are shown in Table V. Larger c results in more training data, which may increase recommendation accuracy at the cost of more training time. Besides, the number of noise items also increases when c is large, which reduces the recommendation performance. Finally, we set the window size c as 3. Moreover, the dimension varies from 50 to 300, and we will explore the optimal value in Section V-C.

All experiments ran on the server with Intel Xeon Silver 4108 CPU, GeForce RTX 2080Ti, 128-GB memory, and Ubuntu 18.04.

TABLE V
EFFECT OF CONTEXT WINDOW SIZE

context window size	Precision@5	Recall@5	F1@5
2	27.76%	7.23%	11.47%
3	**27.96%**	**7.28%**	**11.55%**
4	26.91%	7.01%	11.12%
5	27.09%	7.05%	11.19%
6	27.14%	7.064%	11.21%

B. Visual Illustration of Embedding

We first give the visual illustration of the learned content- and context-aware embeddings (including structure embeddings and text embeddings).

1) Embeddings and Genre: We start with illustrating the structure and text embeddings with t-SNE [59], which provides a visual display of high-dimensional vectors in low-dimensional space. Fig. 5 shows the 2-D structure and text embeddings of several different genres of music pieces.

First, music pieces with similar genres cluster tightly in the visual space. Therefore, both structure and text embeddings learned by CAME can capture music pieces' important features, such as genres and styles, from heterogeneous context and content information effectively. In addition, both structure embeddings and text embeddings reflect some slight differences in genres. For instance, instrumental soundtrack and vocal soundtrack music pieces lie nearby in the 2-D, while they form two close clusters, which also show the effectiveness of the proposed approach CAME. Furthermore, the structure embeddings emphasize relationships between music pieces in the HIN, and the text embeddings lay more importance on the textual content. Therefore, music pieces with common genres tend to have similar text embeddings and generally cluster more tightly.

Besides, the visualization results show the feasibility of applying the embeddings (including the text embeddings and structure embeddings) in various tasks, such as data visualization, music tagging, music retrieval, genre classification, and music clustering.

2) Embeddings and Artists: This section gives the illustration of the structure and text embeddings of top-ten popular music pieces of selected artists with t-SNE. Specifically,

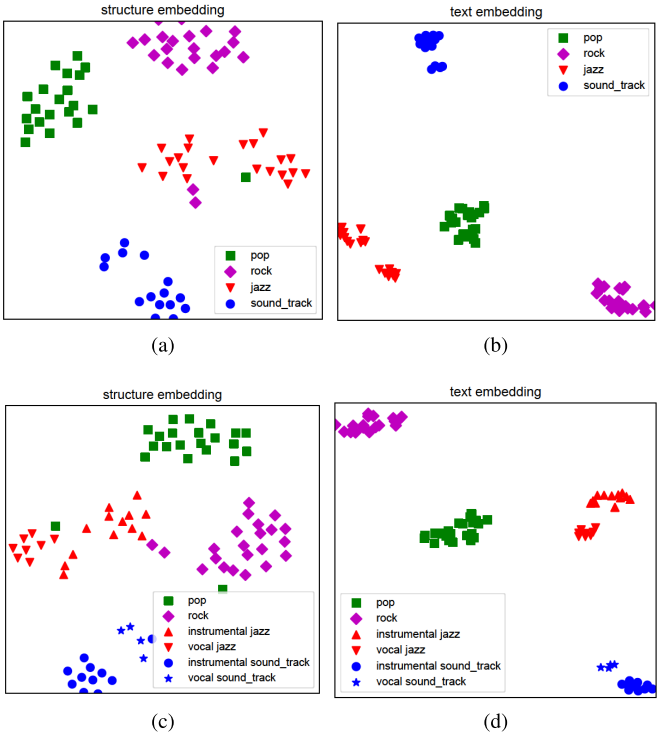

Fig. 5. Visualization of selected music pieces' genre and embeddings in 2-D space. (a) Visualization of structure embeddings, for music pieces with different genres. (b) Visualization of text embeddings for music pieces with different genres. (c) Visualization of structure embeddings for music pieces with different sub-genres. (d) Visualization of text embeddings for music pieces with different sub-genres.

TABLE VI
STYLE INFORMATION OF SELECTED ARTISTS

No.	Artist	Style
1	Guns N' Roses	hard rock, heavy metal
2	Maroon 5	pop, pop rock, funk rock, soul, soft rock
3	Bon Jovi	hard rock, glam metal, arena rock, pop rock
4	Bob Dylan	folk, blues, rock, gospel, country, traditional pop, vocal jazz
5	Robbie Williams	pop rock, soft rock, dance, alternative rock
6	Justin Bieber	pop, R&B, hip hop
7	Lady Gaga	pop, dance, electronic
8	Adele	pop, R&B, soul, blue-eyed soul
9	Mariah Carey	R&B, pop, hip hop, soul
10	Joe Hisaishi	sound track, film score, classical, romanticism
11	Yuki Kajiura	sound track, Japanese, instrumental, anime, j-pop

Table VI gives eleven famous artists and their style information. The 2-D structure embeddings and text embeddings of selected music pieces are shown in Fig. 6.

First, it is interesting to observe that music pieces by the same artist/singer are close in the visual space. The reason is that each artist has her/his own styles, which is also reflected in users' music-listening behaviors as well as the content data, and the proposed model CAME can effectively capture the useful information.

Second, as stated earlier, the structure embeddings emphasize interactions between music and users as well as music

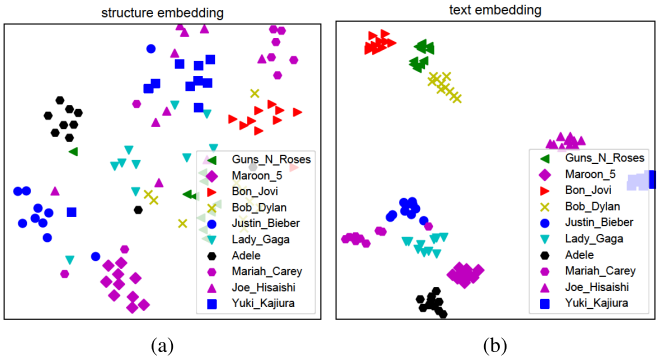

Fig. 6. Visualization of artists' embedding in 2-D space. Visualization of artists' (a) structure embeddings and (b) text embeddings.

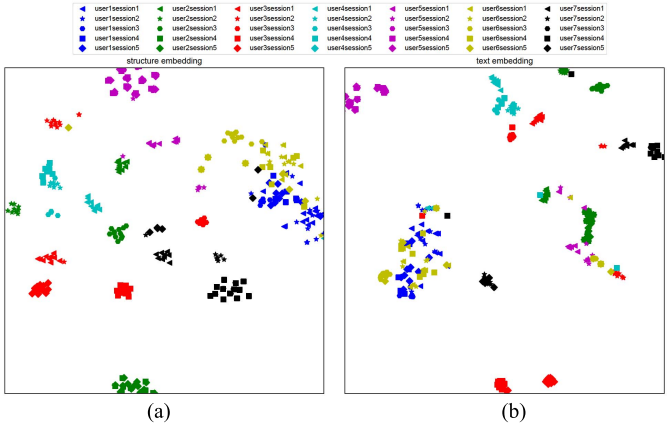

Fig. 7. Visualization of the embeddings of different users' listening records in 2-D space. (a) Visualization of users' structure embeddings. (b) Visualization of users' text embeddings.

playing sequences (context) in the HIN, and the text embeddings mainly depend on the textual content. Therefore, music pieces with the common features tend to have similar text embeddings, which cluster more closely than corresponding structure embeddings.

3) Embeddings and Users: We now give another visual analysis of text and structure embeddings of music pieces in some users' music listening sequences, and the results are shown in Fig. 7.

First, the structure embeddings and text embeddings of music pieces in each user's listening records form one or several clusters, which shows that users generally have specific preferences. For instance, user 1 (blue) prefers only one kind of music, while user 3 (red) likes listening to several kinds of music. Second, the music pieces within the same session generally have similar embeddings, and it shows that every user has more specific contextual preferences that may be different from their general preferences.

In brief, the results show that the structure and text embeddings learned by CAME from context and content information can depict the features of music pieces effectively and also capture their specific aspects as well as users' musical interests adaptively. On the other hand, CAME is also useful for many tasks and applications, such as dimensionality reduction, representation learning, similarity measure, corpus visualization, automatic tagging, and classification.

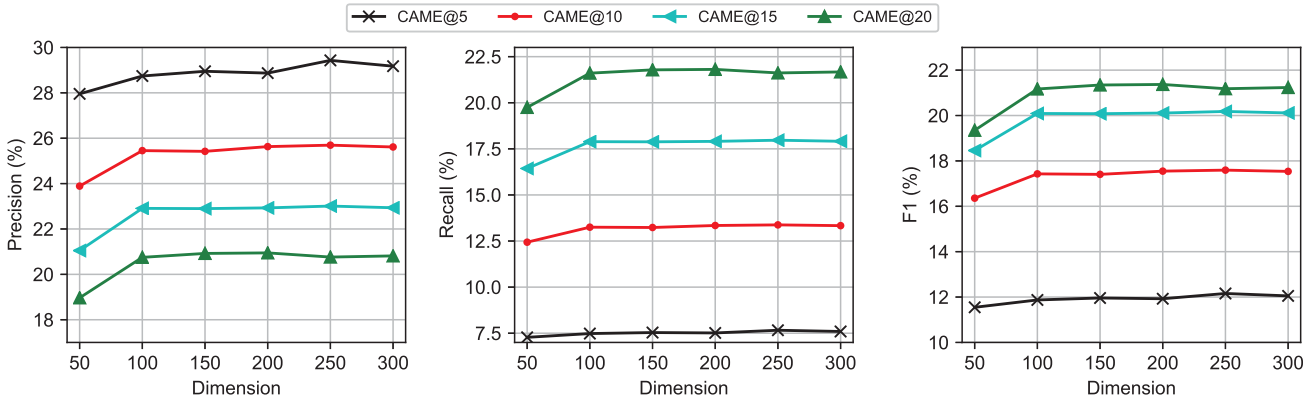

Fig. 8. Experimental results of the dimension's effects.

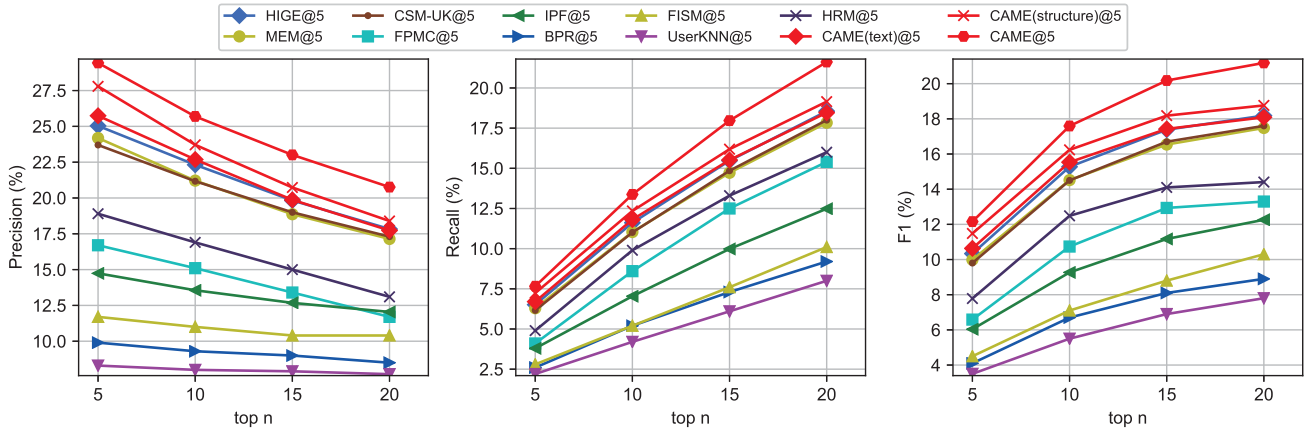

Fig. 9. Performance comparison with baselines.

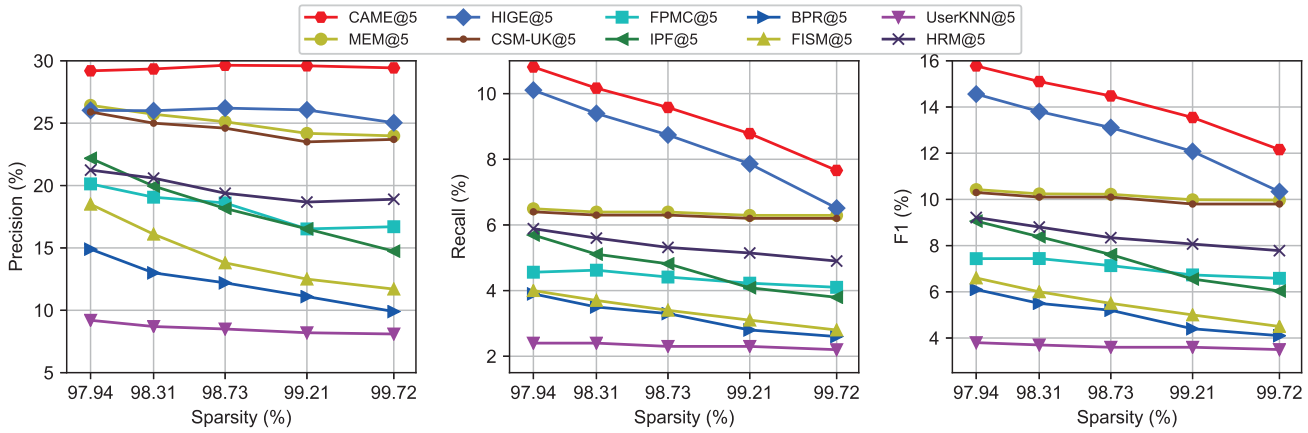

Fig. 10. Performance of $n = 5$ over data sets with a different sparsity.

C. Effects of Dimension

The dimension of the embeddings decides the fitting and modeling ability of the proposed method CAME. Generally, a higher dimension means that CAME can depict more useful information and may have better performance in tasks, such as recommendation. On the other hand, a higher dimension may reduce the efficiency and cause risks of overfitting. Therefore, the proposed approach CAME is evaluated on a different dimension, which increases from 50 to 300, to investigate how the dimension influences the performance of recommendation.

As shown in Fig. 8, the precision tends to increase and then stabilizes as the dimension increases from 50 to 300. The reason is that higher dimensional embeddings are able to incorporate more useful information and represent music pieces more precisely at the expense of inefficiency or overfitting. Besides, CAME with a low dimension (such as 100) achieves a good performance as well. The reason is that the attention mechanism improves CAME's capacity of capturing relevant features of music pieces adaptively. In addition, as shown in Table VII, CAME with a higher dimension is more

TABLE VII
INFLUENCE OF DIMENSION ON EFFICIENCY

Dimension	Training Time/epoch (s)	Testing Time (s)
50	563	164
100	803	184
150	975	204
200	1006	218
250	1207	240
300	1361	269

time-consuming, and we set the dimension to 250 to achieve a balance between accuracy and efficiency in the following experiments.

D. Comparison Against Baselines

The proposed method CAME is evaluated against nine state-of-the-art baselines as well as CAME's variants. The results are given in Fig. 9, and we have the following conclusions.

1) The proposed approach CAME outperforms its variants and baselines in all evaluation metrics, which shows that CAME is able to model users' preferences effectively and perform accurate music recommendation.
2) CAME's variants include CAME (text) and CAME (structure), which only have text and structure components, respectively. Specifically, CAME (structure) has slightly better performance than CAME (text) and CAME outperforms its variants. Therefore, both textual content data and structural context information are important in the recommendation, which can be utilized by CAME effectively.
3) CAME outperforms other embedding-based methods (MEM, HRM, HIGE, and CSM-UK) because it can fully exploit heterogeneous context and content information with CNNs and learn the structure and text embeddings effectively. Especially, the attention mechanism enables CAME to capture the key features of music pieces as well as their dynamic relevance adaptively and guides CAME to emphasize the information that is important in listening behavior modeling and music recommendation.
4) CAME outperforms FPMC because it can learn more important information other than the relationships between adjacent items and make full use of content and context by using network embedding as well as attention mechanism.
5) CAME outperforms BPR, FISM, and UserKNN, and the reason is that it performs recommendations based on both general and contextual preferences, but these baselines only consider users' general preferences.

Therefore, the proposed approach CAME can learn the content- and context-aware embeddings precisely and adaptively and take both general and contextual preferences into consideration to further improve the performance of recommendation.

E. Effects of Data Sparsity

We now explore how the sparsity of data influences the performance of the proposed method CAME as well as the baselines by evaluating all methods on data sets with different sparsity {99.72%, 99.21%, 98.73%, 98.31%, 97.94%}. Specifically, the data sets are generated via filtering music pieces with low listening frequency, which is set as {0, 5, 10, 15, 20}, separately. The results are shown in Fig. 10, and CAME outperforms all baselines over all data sets. The reason is that CAME performs recommendations based on various information, and it can handle sparse data effectively.

VI. CONCLUSION

In this article, we propose a content- and context-aware music recommendation method based on network embedding with attention mechanism and CNNs. Specifically, the proposed method is composed of three components: HIN for incorporating various information, CAME for feature representation learning, and content- and context-aware music recommendation approach. This work differs from previous work in three aspects: 1) the proposed method incorporates and leverages heterogeneous information to alleviate the data sparsity problem; 2) it can cope with various aspects of music when interacting with different neighbors adaptively; and 3) it is able to capture music pieces' dynamic features and learning the structure and text embeddings precisely to further improve the performance of recommendation. Comprehensive experiments, including quantitative and qualitative evaluations, have been performed on real-world music data sets, and we can conclude that: 1) the proposed approach can effectively learn the embedding of music from abundant auxiliary/side information and apply them in recommendation tasks and 2) the context and content information is quite important in achieving accurate personalized recommendation as well as alleviating data sparsity problem.

In the future, we would like to combine advanced techniques [60] with attention mechanisms to incorporate more data, including implicit and explicit feedbacks as well as cross-domain knowledge [61], and extract the key correlations between them to further improve the recommendation results.

REFERENCES

[1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–59, 1997.
[2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.
[3] X. Cai, J. Han, W. Li, R. Zhang, S. Pan, and L. Yang, "A three-layered mutually reinforced model for personalized citation recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6026–6037, Dec. 2018.
[4] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1084–1096, Mar. 2019.
[5] R. Wang *et al.*, "TaxiRec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 585–598, Mar. 2018.
[6] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017.

[7] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 1–34.

[8] D. Lian *et al.*, "Scalable content-aware collaborative filtering for location recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1122–1135, Jun. 2018.

[9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[10] D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. L. Batista, and M. N. M. García, "A collaborative filtering method for music recommendation using playing coefficients for artists and users," *Expert Syst. Appl.*, vol. 66, pp. 234–244, Dec. 2016.

[11] B. R. Cami, H. Hassanpour, and H. Mashayekhi, "User preferences modeling using Dirichlet process mixture model for a content-based recommender system," *Knowl.-Based Syst.*, vol. 163, pp. 644–655, Jan. 2019.

[12] G. Zhong, H. Wang, and W. Jiao, "MusicCNNs: A new benchmark on content-based music recommendation," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 394–405.

[13] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latenttopic sequential patterns," in *Proc. 6th ACM Conf. Rec. Syst.*, 2012, pp. 131–138.

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

[15] L. Xiang *et al.*, "Temporal recommendation on graphs via long-and short-term preference fusion," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 723–732.

[16] M. Kaminskas, F. Ricci, and M. Schedl, "Location-aware music recommendation using auto-tagging and hybrid matching," in *Proc. 7th ACM Conf. Rec. Syst.*, 2013, pp. 17–24.

[17] W.-P. Lee, C.-T. Chen, J.-Y. Huang, and J.-Y. Liang, "A smartphone-based activity-aware system for music streaming recommendation," *Knowl.-Based Syst.*, vol. 131, pp. 70–82, Sep. 2017.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[19] Y. He, C. Wang, and C. Jiang, "Correlated matrix factorization for recommendation with implicit feedback," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 3, pp. 451–464, Mar. 2019.

[20] H. Yang, Y. Zhao, J. Xia, B. Yao, M. Zhang, and K. Zheng, "Music playlist recommendation with long short-term memory," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2019, pp. 416–432.

[21] S. Oramas, V. C. Ostuni, T. D. Noia, X. Serra, and E. D. Sciascio, "Sound and music recommendation with knowledge graphs," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 2, pp. 1–21, Jan. 2017.

[22] Y. Pang, Z. Ji, P. Jing, and X. Li, "Ranking graph embedding for learning to rerank," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1292–1303, Aug. 2013.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[25] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 2111–2117.

[26] N. Passalis and A. Tefas, "Dimensionality reduction using similarity-induced embeddings," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3429–3441, Aug. 2017.

[27] N. Passalis and A. Tefas, "Unsupervised knowledge transfer using similarity embeddings," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 946–950, Mar. 2019.

[28] C. Tu, H. Liu, Z. Liu, and M. Sun, "CANE: Context-aware network embedding for relation modeling," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1722–1731.

[29] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[30] L. Wang, L. Liu, and L. Zhou, "A graph-embedding approach to hierarchical visual word mergence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 308–320, Feb. 2017.

[31] X. Zhang, W. Hu, S. Chen, and S. Maybank, "Graph-embedding-based learning for robust object tracking," *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 1072–1084, Feb. 2014.

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

[33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.

[34] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2016, pp. 207–212.

[35] W. Xu and Y. Tan, "Semisupervised text classification by variational autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 295–308, Jan. 2020.

[36] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, and A. Aït-Bachir, "Period-aware content attention RNNs for time series forecasting with missing values," *Neurocomputing*, vol. 312, pp. 177–186, Oct. 2018.

[37] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1407–1418, May 2019.

[38] W. Pei, J. Yang, Z. Sun, J. Zhang, A. Bozzon, and D. M. J. Tax, "Interacting attention-gated recurrent networks for recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1459–1468.

[39] Y. Miura, M. Taniguchi, T. Taniguchi, and T. Ohkuma, "Unifying text, metadata, and user network representations with a neural network for geolocation prediction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1260–1272.

[40] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 335–344.

[41] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 3119–3125.

[42] X. Wang *et al.*, "Dynamic attention deep model for article recommendation by learning human editors' demonstration," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 2051–2059.

[43] J. Han *et al.*, "Adaptive deep modeling of users and items using side information for recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 737–748, Mar. 2020.

[44] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*. Baltimore, MD, USA: Association for Computational Linguistics, 2014, pp. 655–665.

[45] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1705–1715, Jun. 2019.

[46] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[47] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Inf. Process. Manage.*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[48] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[49] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 891–900.

[50] D. Wang, S. Deng, S. Liu, and G. Xu, "Improving music recommendation using distributed representation," in *Proc. 25th Int. Conf. Companion World Wide Web*, 2016, pp. 125–126.

[51] L. A. Adamic and B. A. Huberman, "Power-law distribution of the world wide Web," *Science*, vol. 287, no. 5461, p. 2115, 2000.

[52] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186.

[53] S. Kabbur, X. Ning, and G. Karypis, "Fism: Factored item similarity models for top-n recommender systems," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 659–667.

[54] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 811–820.

[55] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for NextBasket recommendation," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 403–412.
[56] D. Wang, S. Deng, and G. Xu, "Sequence-based context-aware music recommendation," *Inf. Retr. J.*, vol. 21, nos. 2–3, pp. 230–252, Jun. 2018.
[57] D. Wang, S. Deng, X. Zhang, and G. Xu, "Learning to embed music and metadata for context-aware music recommendation," *World Wide Web*, vol. 21, no. 5, pp. 1399–1423, Sep. 2018.
[58] D. Wang, G. Xu, and S. Deng, "Music recommendation via heterogeneous information graph embedding," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 596–603.
[59] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, nos. 2579–2605, p. 85, 2008.
[60] M. Shi, Y. Tang, and J. Liu, "Functional and contextual attention-based LSTM for service recommendation in mashup creation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1077–1090, May 2019.
[61] Q. Zhang, J. Lu, D. Wu, and G. Zhang, "A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1998–2012, Jul. 2019.

Dongjing Wang received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2012 and 2018, respectively.

He was co-trained at the University of Technology Sydney, Ultimo, NSW, Australia, for one year. He is currently a Lecturer with Hangzhou Dianzi University, Hangzhou. His current research interests include recommender systems, machine learning, and business process management.

Xin Zhang received the bachelor's and Ph.D. degrees in computer science and technology from Shandong University, Jinan, China, in 2012 and 2018, respectively.

She was co-trained at the University of California at Davis, Davis, CA, USA, for one year. She is currently a Lecturer with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. Her research interests include image processing, computer vision, and machine learning.

Dongjin Yu (Member, IEEE) is currently a Professor with Hangzhou Dianzi University, Hangzhou, China, where he is also the Director of the Institute of Big Data and the Institute of Computer Software. His research efforts include big data, business process management, and software engineering.

Dr. Yu is also a member of ACM and a Senior Member of the China Computer Federation (CCF). He is also a member of the Technical Committee of Software Engineering of CCF and the Technical Committee of Service Computing of CCF.

Guandong Xu (Member, IEEE) received the Ph.D. degree in computer science from the School of Computer Science and the Advanced Analytics Institute, University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Full Professor of data science with the School of Computer Science and the Advanced Analytics Institute, University of Technology Sydney. He has published three monographs in Springer and CRC press and more than 190 journal articles and conference papers, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON SERVICES COMPUTING (TSC), the International Joint Conference on Artificial Intelligence (IJCAI), the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, the International World Wide Web Conference (WWW), the IEEE International Conference on Data Engineering (ICDE), and the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) conferences. His research interests cover data science, data analytics, recommender systems, web mining, user modeling, Natural Language Processing (NLP), social network analysis, and social media mining.

Dr. Xu received a number of Industry Awards from the Australian Industry Community, such as the 2018 Top-10 Australian Analytics Leader Award. He is also the Assistant Editor-in-Chief of the *World Wide Web* journal. He has been serving on the editorial board or as a guest editor for several international journals.

Shuiguang Deng (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

He was a Visiting Scholar with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2014, and Stanford University, Stanford, CA, USA, in 2015. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He has published over 80 articles in journals, such as the IEEE TRANSACTIONS ON COMPUTERS (TOC), the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), the IEEE TRANSACTIONS ON SERVICES COMPUTING (TSC), the IEEE TRANSACTIONS ON CYBERNETICS (TCYB), and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and refereed conferences. His research interests include service computing, mobile computing, and business process management.

Dr. Deng is an Associate Editor of IEEE ACCESS and the *IET Cyber-Physical Systems: Theory & Applications*.