Personalized and Context-Aware Music Retrieval and Recommendation

ZHIYONG CHENG

SINGAPORE MANAGEMENT UNIVERSITY

2016

Personalized and Context-Aware Music Retrieval and Recommendation

by

Zhiyong Cheng

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Jialie Shen (Supervisor/Chair)
Assistant Professor of Information Systems
Singapore Management University

Shuicheng Yan (Supervisor)
Associate Professor of ECE Department
National University of Singapore

Steven C.H. Hoi
Associate Professor of Information Systems
Singapore Management University

Feida Zhu
Assistant Professor of Computer Science
Singapore Management University

Singapore Management University
2016

Personalized and Context-Aware Music Retrieval and Recommendation

by

Zhiyong Cheng

Abstract

Rapid advances in mobile devices and cloud-based music services have brought about a fundamental change in the way people consume music. Cloud-based music streaming platforms like Pandora and Last.fm host an increasing huge volume of music contents. Meanwhile, the ubiquity of wireless infrastructure and advanced mobile devices enable users to access such abundant music content anytime and anywhere. Consequently, there has been an increasing demand for the development of intelligent techniques to facilitate personalized and context-aware music retrieval and recommendation. Most of existing music retrieval systems have not considered users' music preferences, and traditional music recommender systems have not considered the influence of local contexts. As a result, search and recommendation results may not best fit users' music preference influenced by the dynamically changed contexts, when users listen to music using mobile devices on the move. Current mobile devices are equipped with various sensors and typically for personal use. Thus, rich user information (e.g., age, gender, listening logs) and various types of contexts (e.g., time. location) can be obtained and detected with the mobile devices, which provide an opportunity to develop personalized and context-aware music retrieval and recommender systems.

Among various contexts that have influences on users' music preferences, venue is a very important one and can be accurately detected by current techniques. Different venues not only have unique background environments and atmosphere, but also highly correlate with local activities and events. These factors play critical roles in determining users' music selections. In the first

work, we develop a venue-aware music recommender (VAMR) system called VenueMusic, which can automatically recommend suitable music tracks to various types of venues. A location-aware topic model is proposed to mine the common features of songs that are suitable for a venue type and map the songs and venue types into the same semantic space. Experimental results demonstrate the effectiveness of VenueMusic and advantages over other systems. In the second work, we develop a user information aware (UIA) user-aware music retrieval system, which can utilize users' demographic information (e.g., age and gender) in text-based retrieval. A UIA music interest topic model is proposed to capture the influence of age and gender on music preferences. Based on this model, a novel UIA retrieval method is proposed. Empirical studies demonstrate that with this method, the performance of various text-based retrieval methods can be significantly improved. As demographic information is not difficult to obtain, the system can be used to deal with the new users for the personalized music retrieval (PMR) system presented in the third work. In this work, a novel dual-layer music preference topic model is proposed to characterize the correlations and interplays between users, songs, and terms under two latent semantic spaces. Comprehensive experiments have been conducted and demonstrate that the PMR system can better satisfy users' music preference and significantly improve the personalized search accuracy.

In summary, we present three music information retrieval systems, which can be integrated together to help users find their favorite music at different venues. Our studies in this thesis are early attempts in the development of user-centric music retrieval systems, especially on the PMR and VAMR. We hope this work can shed light on the direction of developing user-centric music retrieval systems and motivate more studies on this promising research area.

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to offer my sincere and deepest gratitude to my advisor Professor Jialie Shen. His encouragement, supervision, and support enabled me to grow up as a Ph.D. for independently carrying out research. I benefited a lot from his profound knowledge and rigorous attitude toward scientific research. Without his guidance and persistent help, this thesis would never have happened. Also, I am greatly indebted to Dr. Haiyan Miao, for her kindness and continuous concern on my research and life throughout these years.

I would like to thank my thesis committee members, Professor Shuicheng Yan, Professor Steven C.H. Hoi and Professor Feida Zhu. Their comments and suggestions are very constructive for improving this thesis.

I acknowledge the financial, academic and technical support of Singapore Management University. In addition, I thank Professor Ee-Peng Lim, Professor Steven Miller and Professor Stephen E. Fienberg, who provided me with a great opportunity to visit Carnegie Mellon University (CMU). I am very grateful to Professor Alexander Hauptmann for providing me with the opportunity to visit his group Informedia in CMU.

I give my thanks to my family for their everlasting love, patience and support. Finally, I thank my wife, Tian, for her love, support, encouragement, and companionship throughout my Ph.D.

List of Notations

s	A song document
u	A user
v	A audio term in the audio word vocabulary
t	A text term in the text word vocabulary
D	A corpus with user information and music information
$\|D_u\|$	The total number of songs in user u's music profile
v_s, w_s	A audio word and a text word in a song, respectively
$\boldsymbol{v_s}, \boldsymbol{w_s}$	Audio word sequence and text word sequence of a song, respectively
$\|\boldsymbol{v_s}\|, \|\boldsymbol{w_s}\|$	Number of audio words and number of text word in a song
a, g, l	A age category, a gender category, and a venue type, respectively
L	Total number of venues
$\mathcal{U}$, $\mathcal{S}$	User set and song set in the corpus, respectively
U, S	Total number of users and songs in the corpus, respectively
$\mathcal{T}$, $\mathcal{V}$	Text word vocabulary and the audio word vocabulary, respectively
T, V	Vocabulary size of text words and audio words, respectively
$\mathcal{A}$, $\mathcal{G}$	Age category set and gender category set, respectively
A, G	Total number of age groups and gender groups, respectively

z	A latent topic
υ	A latent music dimension
y	An indicator variable
π	A parameter of Bernoulli distribution
$\boldsymbol{\eta}$	Beta priors
$\boldsymbol{\lambda}$	Mixing weight vector of user, age and gender music preference
K	Number of latent topics
M	Number of latent music dimensions
$\boldsymbol{\theta_x}$	Multinomial distribution over latent music dimensions or topics specific to x (e.g., user, venue, or age)
$\boldsymbol{\phi_x}$	Multinomial distribution over x (e.g., songs, audio words, or text words)
N_x^k	Number of times observing topic k in x (e.g., song, venue, and age)
N_k^x	Number of times observing x (e.g., song, audio term or text term) in the topic k
N_m^s	Number of times observing song s in the latent music dimension m
N_m^k	Number of times observing the latent topic k in the latent music dimension m
N_u^m	Number of times observing the latent music dimension m in u's profile
$\boldsymbol{W}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$	Vectors for words, indicators and topics, respectively
$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$	Dirichlet priors

Chapter 1

Introduction

Over the past decades, empowered by fast advances in digital storage and networking, we have witnessed ever-increasing amount of music data. Meanwhile, rapid advances in mobile devices (e.g., mobile phones) and cloud-based music streaming services, such as Last.fm[1] and Spotify[2], have brought about a fundamental change in the way people consume music. Mobile devices become the mainstream platforms allowing people to enjoy favorite music anytime and anywhere. According to Nelsen's Music 360 2015 report, 44% of US music listeners use smartphones to listen to music in a typical week. While large-scale music data available from various sources and fast technical advancements provide users great flexibility and convenience in consuming music, it also introduces an impending and challenging problem about how to assist users in *finding their favorite music* or *satisfying users' music needs* under *dynamically changed contexts* from *large-scale music datasets*.

Music retrieval and recommender systems are two most important tools to enable users to explore large-scale music collections or find favorite music. Music information retrieval system requires users to input a query[3] to represent

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

[2] https://www.spotify.com/

[3] Typically, a query could be in the form of *text words*, such as *artist*, *titles*, *lyrics* or *other annotated descriptive terms* (e.g., genre, mood and instrument), or a piece of music [156] or humming melody [111].

their current music needs, and then estimate the relevance of music items with respect to the query. The most relevant ones are returned to the users. Alternatively, music recommendation systems infer users' music preferences according to their past listening behaviors, find and return music tracks which best fit their music preferences. When seeking music, users aim to find music items which satisfy their music needs or preferences. Therefore, understanding and representing users' music needs are the prerequisite for both types of systems. Users' music needs are dependent on their music taste and preference [4], also called long-term and short-term music preference in this thesis. The long-term music preference relates to users' *personality*, *self-views*, *cognitive ability*, *gender*, and *culture background*, etc; and the short-term music preference relates to users' local *surrounding environment and atmosphere* (e.g., location, time, temperature, ambient lighting conditions, weather, noise level) and physical state (e.g., activity and mood) [126]. There has been a long history in the development of music retrieval and recommender systems [28, 108, 140, 142, 153], however, these systems cannot well satisfy users' increasing requirements on music services. One of the main problems is that existing systems cannot comprehensively understand users' music needs under local contexts, which requires the consideration of both long-term and short-term music preferences.

The *input query* is the most basic form to represent users' music information needs. In general, the input query can only reflect users' short-term music preferences. As a result, the users' long-term music preferences are often ignored. Different users have a wide range of music preferences. Thus, given the same query, different users prefer different results. However, most of the existing music retrieval systems will return the same results, which are not optimal for individuals. Taking a simple example, a query of "sad" represents

[4]Based on the definition of Schedl et al. [136]: " *Taste refers to a long-term inclination and preference describes a rather short-term, situation-dependent affection. Both are likely to change over time, although taste usually changes only gradually and at a slower rate than preference.*".

that the user wants to listen to sad music now. Without the knowledge of the users' long-term music preferences, what type of sad music is suitable for the user is unknown. Consequently, the results could be unsatisfactory for this user. To improve the performance of music retrieval systems, it is crucial to consider users' long-term music preferences in retrieval. In contrast, most of existing music recommender systems only capture users' long-term music preferences while ignoring users' short-term music preferences, which can be greatly influenced by local contexts [126], such as local social activities/events or geo-location. A typical example is that a user may prefer energetic music in the gym while peaceful music in the library. In recent years, context-aware music recommender systems (CAMR) have been attracting increasing attentions. CAMR systems consider the influence of local contexts on users' music preference, and thus could recommend music tracks to better fit users' local music preferences. Despite the high potential of CAMR systems, few CAMR systems have been used in real applications, due to many challenges faced when developing effective CAMR systems. The majority of these challenges pertains to the heterogeneity of data, including complex music content and various types of contexts (e.g., time, location, weather, activity, and mood). Another big challenge is related to context-aware system evaluation - the lack of standard test collections and system performance assessment framework makes every evaluation time-consuming and often requires real users' judgments [126].

1.1 Thesis Focus, Research Challenges and Main Contributions

The advanced mobile devices, ubiquitous wireless infrastructure, and cloud-based music streaming services enable general users to access large-scale music contents anytime and anywhere. It provides users great convenience to enjoy

abundant music contents. At the same time, it also brings new requirements and challenges for the development of music information retrieval systems. The mobile music listening platform requires that the music retrieval and recommender systems can effectively identify and retrieve users' favorite songs under the dynamically changed contexts when on the move. Therefore, it is important but challenging to develop effective personalized and context-aware music retrieval/recommender systems. On the other hand, current mobile devices, such as smartphones, are embedded with different types of sensors (i.e., GPS, camera, microphones, gyroscopes, ambient light sensors). Thus, various types of contextual information can be detected and collected, such as time, location, weather, ambient light and sounds. Besides, mobile devices are typically for personal use. Therefore, it is not difficult to obtain users' personal information (e.g., age, gender and listening logs). The availability of such personal and contextual information facilitates the development of personalized and context-aware music systems and applications.

As discussed, most of the existing music retrieval systems have not considered users' long-term music preferences and traditional music recommender systems ignore the influence of contextual factors on short-term music preferences. Thus, it is important to develop *personalized music retrieval (PMR) systems*, which take users' long-term music preferences into consideration, and *context-aware music recommender (CAMR) systems*, which consider the influence of contextual factors. Specifically, in this thesis, main research focus is on the development of venue-aware music recommender (VAMR) and personalized text-based music retrieval (PTBMR) systems. The specific motivations are,

- For venue-aware music recommendation. Location is a very important context which significantly affects users' music preferences [19]. However, studies on location context in music recommendation are very sparse.

Particularly, venue, as an important location context, which is not only directly related to the surrounding atmosphere but also related to the local activities and events, has not been considered in previous context-aware music recommender systems.

- For personalized text-based music retrieval. Firstly, text-based music retrieval (TBMR) is a natural and easy way for users to use and has been widely used in existing music services, such as Last.fm and Youtube[5]. Besides, current TBMR systems have not considered users' music preferences and retrieve songs only based on their relevances with respect to the query. Given a query, the search results could be very poor for some users. PTBMR systems could evaluate the relevance of songs, according to *users' personal preferences on the song with respect to a query*. Despite the potential of PTBMR systems, few such systems have been developed.

The VAMR system aims to automatically recommend suitable songs for different venue types, such as recommending music tracks for gym and library. The PTBMR system is developed to provide personalized search results. Notice that the two systems can be integrated together to provide music services. For instance, when a user arrives at a particular venue, the VAMR system automatically recommends music tracks for this venue. In the case that the results do not fit her current music preferences because of other factors (e.g., mood), the user can express her current music needs with semantic concepts (e.g., happy). Then the PTBMR system will refine the recommended results and return personalized results to fit her current preferences at this venue.

Although PMR and CAMR have attracted increasing research attentions in recent years [89, 133, 136, 126], the development of related systems is still in the early stage. Many challenges have not been properly addressed. As far

[5]https://www.youtube.com/

as this thesis concerned, we confront the following research challenges.

- **Challenges on data complexity:** The data complexity mainly comes from two aspects - (1) the complexity of music content: the analysis of music content is crucial for both music retrieval and recommendation. Due to the well-known "semantic gap" - the gap between high-level concepts (e.g., genre and mood) used by human to interpret the music and the low-level acoustic features used by computers to describe the audio stream [126], the performance of content-based music retrieval and recommendation is still far from satisfactory. Consequently, text-based music methods and collaborative filtering [131] techniques are dominant in music retrieval and music recommendation systems, respectively; and (2) data heterogeneity: in addition to the complexity of music content, researchers need to consider many different types of data closely around the music: associated textual data (e.g., audio, metadata, tag or annotation), users' related data (e.g., age, gender, listening logs) and various types of contextual data (e.g., location, time, emotions). High data complexity also increases the difficulty in datasets construction and music preference modeling.

- **Challenges on constructing data collection:** In the development of PMR and CAMR systems, datasets are the foundation to analyze and model users' music preference and the influence of contextual factors, as well as system evaluation. With more types of data involved, it becomes more difficult to collect data. With the popularity of social music websites (e.g., Last.fm), it is relatively easy to collect large-scale music data (e.g., tags), users' profile and their listening logs (e.g., when and how many times a user listens to a song). However, it is very difficult to obtain the contexts under which the user likes/listens to a song. As a result, researchers have to conduct a user study to collect the data for

their studies. Because of the high cost of time and labor consuming, the collected datasets is rather small, as can be seen in Table 2.1.

- **Challenges on modeling users' (contextual) music preferences:** Users' (contextual) music preferences modeling is the key to the success of personal music retrieval (and context-aware music recommendation). Human perceives and judges music based on the high-level semantics (such as emotion, mood and genre) embedded in music contents. However, music semantic meanings cannot be effectively characterized using low-level spectral features due to the "semantic gap" [169]. Thus, in traditional music recommendation, collaborative filtering [131] is often used to avoid dealing with music contents. Unfortunately, this technique cannot be used in PMR. Because the results not only need to match users' music preference but also are relevant to the query. It implies that we have to deal with the music contents to capture the associations between (user, song, content). Similarly, CAMR needs to model users' music preference under different contexts, namely, capturing the associations between (song, context) for general CAMR (to recommend music to certain contexts) or (user, song, context) for personalized CAMR. Because the data sparsity [126] becomes much severer in CAMR (as it is more difficult to get data of users under different contexts), collaborative filtering methods usually cannot obtain good performance. As a result, CAMR also has to face the problem of analyzing music content to capture the complex interactions between (song, content, context) or even (user, song, content, context) for personalized CAMR.

- **Challenges on system evaluation:** System performance evaluation is crucial for the development of any information retrieval system. How to comprehensively and fairly evaluate a retrieval system has been an important research topic in information retrieval. Evaluating PMR and

CAMR systems is lack of reference datasets and evaluation frameworks. The evaluation of such systems often requires the participation of real users. However, it is very expensive in terms of both labor and time for conducting large-scale user study. Previous researches generally limit to a small group of participants (as shown in Table 2.1). The problem is that results obtained based on a small population are easy to be biased. How to comprehensively and fairly evaluate and compare PMR and CAMR systems in a reproducible setting is very difficult.

In this thesis, we develop a VAMR system and a PTBMR system. Because the PTBMR system suffers the cold start problem of new users[6], we also develop a user information aware text-based music retrieval system to relieve the problem. In this system, users' demographic information (e.g., age and gender) are used in retrieval to improve the search results. In the development and evaluation of these systems, we make following main contributions.

1. Three personalized and context-aware music retrieval and recommender systems are developed. In the thesis, we develop three systems: (1) a venue-aware music recommender system - recommends suitable music tracks to different venue types, (2) user information aware text-based music retrieval (UIA-TBMR)system - utilizes users' demographic information, such as age and gender, in text-based music retrieval and can significantly improve the search accuracy, and (3) a personalized text-based music retrieval (PTBMR) system - captures and utilizes users' music preferences in text-based music retrieval to improve music search accuracy. No music retrieval and recommender system with the same functionality as our proposed systems has been reported in previous literatures. Notice that the PTBMR system requires the listening records of users to learn their music preferences, while the UIA-TBMR system only

[6]The system does not have users' music data to learn their music preferences.

needs users' demographic information, which is easy to obtain. Thus, the two systems are complimentary: the UIA-TBMR system could be used for users with no or few listening records, and the PTBMR system is used for users with enough listening records.

2. We propose latent topic models to capture users' music preferences in personalized text-based music retrieval and context-aware music recommendation in the latent semantic space. To address the challenges on music preference modeling in PMR and CAMR, we use both music semantic concepts and music acoustic features to construct a latent music space using topic modeling methods. Both songs and the music preferences of contexts and users are mapped into the latent music space. Thus, songs and music preferences (of contexts and users) are represented by the same latent topics and can be directly matched. In Chapter 3, a location-aware topic model is proposed to represent both venues and songs as the probabilistic distributions of the same set of latent topics. In Chapter 4, we propose a User Information Aware Music Interest Topic (UIA-MIT) Model to capture the general effects of gender and age on the music preference of users in a latent semantic space. In Chapter 5, a novel Dual Layer Music Preference Topic Model is proposed to construct a latent music interest space and characterize the correlations and interplays between users, songs, and keywords or terms under the latent space.

3. We construct several large-scale datasets for system evaluation. For each proposed system, we have conducted a set of experiments to evaluate its effectiveness and compare with related methods and systems. Datasets and evaluation methodologies are two key components for system evaluation. To construct the data collections for performance evaluation, large-scale user related data (e.g., age, gender, and listening log) and

music related data (e.g., title, tags, and audio track) are collected from several social platforms, such as Last.fm, Grooveshark[7], Spotify, and Twitter[8], as well as other expert-based music websites, like 7digital[9] and Allmusic[10]. The data collected from multiple platforms are processed and merged to constructed training and testing datasets for evaluating the proposed systems. In evaluation, we design offline experiments on held-out data and online experiments by user study to compare with a set of competitors. The advantage of offline experiments is its scalability and reproducibility. However, the offline experiments cannot evaluate the performance of the systems in real scenarios, as the evaluated performances are based on the labeled data, which is usually not complete. On the other hand, the online user study can evaluate the effectiveness of systems in real scenarios, while it is usually in small scale. Therefore, we use both offline and online experiments to comprehensively evaluate the developed systems.

1.2 Structure of the Thesis

In Chapter 2, we firstly briefly introduce music retrieval and recommendation techniques and topic models, and then comprehensively review context-aware music recommender systems in literature. In Chapter 3 - Chapter 5, we present the developed venue-aware music recommender system, UIA-TBMR system and PTBMR system, respectively. Finally, we conclude the thesis and discuss promising directions of future work.

[7] http://grooveshark.com/
[8] https://twitter.com/
[9] https://www.7digital.com/
[10] http://www.allmusic.com/

1.3 Origins and Publications

The following publications form the basis of chapters in this thesis:

- Chapter 3:
  - **Z. Cheng** and J. Shen, On effective location-aware music recommendation, *ACM Transactions on Information Systems (TOIS)*, 2016 (to appear).
  - **Z. Cheng** and J. Shen, VenueMusic: a venue-aware music recommender system, In *Proceedings of the International ACM SIGIR Conference on Research & Development on Information Retrieval (ACM SIGIR)*, 2015 (demo paper).
- Chapter 4:
  - **Z. Cheng** and J. Shen, Exploring user-specific information in semantic-based music retrieval, *submitted to TOIS*, (Major revision).
- Chapter 5:
  - **Z. Cheng**, J. Shen and S. Hoi, On effective personalized music retrieval via exploring online user behaviors, In *Proceedings of the International ACM SIGIR Conference on Research & Development on Information Retrieval (ACM SIGIR)*, 2016 (accepted as full paper).

The following are the papers published during the course of the Ph.D but not included in this thesis:

- **Z. Cheng**, X. Li, J. Shen and A. Hauptmann, Which Information Sources are More Effective and Reliable in Video Search, In *Proceedings of the International ACM SIGIR Conference on Research & Development on Information Retrieval (ACM SIGIR)*, 2016 (short paper).

- **Z. Cheng** and J. Shen, Just-for-Me: an adaptive personalization system for location-aware social music recommendation, In *Proceedings of the ACM International Conference on Multimedia Information Retrieval (ACM ICMR)*, 2014.

- **Z. Cheng**, J. Shen and T. Mei, Just-for-Me: an adaptive personalization system for location-aware social music recommendation, In *Proceedings of the International ACM SIGIR Conference on Research & Development on Information Retrieval (ACM SIGIR)*, 2014 (demo paper).

- **Z. Cheng**, X. Li, J. Shen and A. G. Hauptmann, CMU-SMU@TRECVID 2015: Video Hyperlinking, *TRECVID 2015 Video Hyperlinking Competition*, (1st place in MAP and 2nd place in MaiSP, to appear).

- **Z. Cheng**, and J. Shen, On very large scale test collection for landmark image search benchmarking, *Signal Processing*, available online[11].

- **Z. Cheng**, J. Shen, and H. Miao, The effects of multiple query evidences on social image retrieval systems, *ACM Multimedia Systems Journal*, available online[12].

- J. Shen, R. H. Deng, **Z. Cheng**, L. Nie, and S. Yan, On robust image spam filtering via comprehensive visual modeling, *Pattern Recognition*, 48(10): 3227-3238.

- J. Shen, **Z. Cheng**, J. Shen, T. Mei, and X. Gao, The evolution of research on multimedia travel guide search and recommender systems, In *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2014.

- J. Ren, **Z. Cheng**, J. Shen, and F. Zhu, Influence of influential users: an empirical study of music social networks, In *Proceedings of the Interna-*

[11]http://www.sciencedirect.com/science/article/pii/S0165168415003813
[12]http://link.springer.com/article/10.1007/s00530-014-0432-7/fulltext.html

tional Conference on Internet Multimedia Computing and Service (ACM ICIMCS), 2014.

- **Z. Cheng**, J. Ren, J. Shen, and H. Miao, Building a large scale test collection for effective benchmarking of mobile landmark search, In *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2013.
- **Z. Cheng**, J. Ren, J. Shen and H. Miao, The effects of heterogeneous information combination on large scale social image search, In *Proceedings of the International Conference on Internet Multimedia Computing and Service (ACM ICIMCS)*, 2011.

Chapter 2

Related Work

In this chapter, we briefly review the background knowledge and techniques that are closely related to our research topics and methods in the following chapters. We start with an introduction of music information retrieval in Section 2.1, where main focus is on text-based music retrieval, as well as personalized music retrieval. In Section 2.2, we introduce the general techniques in music recommender systems and review the research development in the domain of context-aware music recommendation. As music retrieval models introduced in Chapter 3 - Chapter 5 are based on topic modeling, in Section 2.3, we briefly recall related latent topic models and discuss the differences of our proposed topic models.

2.1 Music Information Retrieval

Music information retrieval is an important sub-domain of information retrieval with many real applications. The general method of information retrieval is to estimate the relevance or similarity between the query and documents in the dataset, and return the most relevant ones. Broadly, there are two music retrieval paradigms in music retrieval: one is based on the features extracted from the audio signal of the music tracks, called content-based music retrieval;

and the other one is based on the associated texts (e..g, artist, title, lyrics and other semantic musical concepts - genre, mood, instruments, etc.), called text-based music retrieval. In this section, we mainly focus on the text-based music retrieval, because text-based music retrieval is more popular in real applications [27] and our studies presented in Chapter 4 and Chapter 5 are based on text-based music retrieval.

2.1.1 Content-based Music Retrieval

Content-based music retrieval systems extract acoustic features from the sound tracks of music and then compute the similarity between music items based on the acoustic features. Thus, *music feature extraction* and *music similarity measurement* are the core components in content-based music retrieval systems. Many types of music features have been proposed and used, such as timbre, pitch, melody, chroma, and rhythm features. Comprehensive introduction about these features can be found in [27, 134]. How to develop effective music similarity measurement is a very active topic of research in music retrieval and recommendation. The main research problem in music similarity is to define a suitable distance or similarity measures - compute the distance between two music tracks, such as Euclidean, Manhattan and Hamming distance [25]. Suitable distance measurement is highly dependent on the used features. For example, when representing music tracks as Gaussian Mixture Models (GMMs) of mel-frequency cepstral coefficients(MFCCs), Kullback-Leibler distance and Earth Mover Distance could achieve much better performance than Euclidean distance [58].

Query by example (QBE) and query by humming (QBH) are two typical paradigms of content-based music retrieval. QBE takes a piece of music track as an input and returns the metadata information of the recording - artist, title, etc. A typical scenario is that a user wants to obtain the metadata information

of an unknown track. Other applications of QBE include plagiarism detection, copyright monitoring, etc. In QBH systems, the input is a melody sung by the user and retrieves the matching track and its metadata. QBH could be used in the scenarios that the user can only hum melodies that are memorable while there is no record of music track at hand.

Despite the advances of content-based music retrieval research, content-based music retrieval systems still fail to cover the semantic distance between high-level semantic concepts (the language used by human) and low-level features. Researchers in the field of music information retrieval suggest to find algorithms for representing music at a higher, more conceptual abstraction level. For more related works about content-based music retrieval, please refer to [153, 27].

2.1.2 Text-based Music Retrieval

Traditional text-based music retrieval techniques heavily rely on the meta-information (e.g., artist and title) and well-defined categorized information (e.g., genre and instrument). In many cases, users would also like to describe their current contexts, such as *emotions* and *occasions or filmed events*, with the expectation that the music search engines return a playlist with suitable songs [68]. To support the search of such semantic queries, it needs to annotate songs with a rich vocabulary of music terms, which requires musical expert knowledge. A typical example is Pandora[1], which relies on experts to generate description on songs. However, expert-based annotation is very time-consuming and labor expensive and thus unlikely to scale with the growth in the amount of recorded songs.

To deal with the problem, many auto-tagging methods are proposed to automatically annotate songs with music related tags/terms by learning the correlation between music acoustic contents and the semantic terms based on a

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

well-annotated music collection. Most auto-tagging systems generate a vector of tag weights when annotating a new song for music search and retrieval [100]. An early work in this direction was performed by Turnbull et al. [151]. They formalize the audio annotation and retrieval as supervised, multi-class labeling task. The dataset CAL500 they created for this study becomes the standard test collections for subsequence works [152, 158, 41, 100, 102, 8]. Miotto et al. [100] proposed a generative approach to improve automatic music annotation by modeling contextual relationships between tags. Recently, Ellis et al. [41] proposed to use a higher-level "Bag-of-Systems" (BoS) representation of the characteristics of music piece to improve the auto-tagging performance.

With the advent of social music websites, songs are annotated with user-contributed social tags, which provide an alternative way to navigate and search songs (e.g., Last.fm). Social tags, which are contributed by a community of internet users and have no constraints on the use of text, provide a rich vocabulary and cover most terms usually used to describe songs. Extensive research efforts have been devoted into developing tag-based music search systems [76, 86, 87]. However, the user-provided tags are known to be noisy, incomplete and subjective [87], which limit the search performance of tag-based methods. Consequently, many works consider the combination of tags and acoustic similarity to improve the search performance [87, 101, 70, 71, 69]. For example, Levy et al. [87] represented a music track with a joint vocabulary consisting of social tags and muswords, and then apply text-based information retrieval techniques to music collections; Knees et al. [70] incorporated audio-based similarity into a tag-based ranking process, either by directly modifying the retrieval process or by performing post-hoc audio-based reranking of the search results; Miotto et al. [101] combined tags and acoustic contents in retrieval via a probabilistic graph-based representation.

2.1.3 Personalized Music Information Retrieval

The works mentioned above measure the relevance of a music track with respect to a query only based on the similarity between the query's content (textual or/and acoustic contents) and the track's content, while ignoring user's music preference. Actually, the perception on music is very subjective to personal preference. Therefore, users' opinions on the search results with respect to a query could be very different. In recent years, researchers have emphasized the importance of considering user's information in music retrieval and advocated to develop user-centered music retrieval systems [89, 133].

In the domain of text retrieval, personalized information retrieval has attracted lots of research attentions, and many approaches have been proposed in last decades [23, 47, 94, 141, 162]. However, very few works have been reported for personalized text-based music retrieval. Hoashi et al. [51] used relevance feedback methods to refine users profiles for improving search performance, while the method was designed for content-based music retrieval systems. Wang et al. [158] proposed a tag query interface which enables users to specify their query in multiple tags and with multiple levels of preferences. This method relies on user's efforts to specify the importance of query tags in each query session. Symeonidis et al. [148] proposed to apply the high order singular value decomposition (SVD) method to capture the associations between $(user, tag, item)$. Based on the likeliness that user u will tag musical item i with tag t, they recommend musical items to user u. However, this method suffers from the high time complexity of SVD and is only applicable for small scale data. Hariri et al. [48] considered the problem of personalized text-based music retrieval where users' history of preferences are taken into account in addition to their issued textual queries. They used music annotations retrieved from social tagging Websites such as Last.fm and use them as textual descriptions of songs. However, they have not evaluated the proposed

system in retrieval evaluation framework and have not compared with other music retrieval methods. In Chapter 5, we present a novel personalized text-based music retrieval system, which uses a dual-layer topic model to explore the correlations among (user, song, term) for retrieval. Besides, we evaluate the system in ad-hoc retrieval tasks on two test collections, and compare it with other text-based music retrieval methods. Experimental results show that the system can significantly improve the search accuracy with respect to personal preferences.

2.2 Music Recommendation

Music recommendation can be dated back to 1994 [139], not much later than the born of the field of recommender systems in the early 90's [126]. However, the major breakthrough came around at the turn of 2000's, when large amount of music contents became available and online music services provide convenient channels for people to access the online music contents, which create large music communities and allow major music recommender systems to emerge. Music recommendation is a challenging task not only because of the complexity of music content, but also because human perception of music is still not thoroughly understood [126], which can be influenced by age, gender, personality traits, cultural background, and other contextual factors.

In recent years, due to the ubiquity of wireless and advanced mobile devices, which can detect surrounding contexts and enable users to access music contents anytime and anywhere, context-aware music recommender (CAMR) systems has emerged and become a hot research field. The idea is to recommend music depending on the user's actual situation, such as her activity or emotional state, or any other contextual factors which might influence the user's local music preferences. For example, location-aware music recommender systems can find music tracks that match the atmosphere of the user's

location. Despite the high potential of the applications, the development of real-world CAMR systems is still in its early stages, because of (1) the heterogeneity of data - researchers have to deal with the complex music content and various types of contextual information (e.g., emotion, time, and location) and (2) the high cost of evaluating context-aware systems - the lack of datasets and evaluation frameworks makes the evaluation very difficult and often requires user study.

This section reviews the music recommendation techniques and the current state of CAMR systems.

2.2.1 General Techniques

Existing music recommender systems can be generally classified into three different categories: collaborative filtering, content-based and hybrid-based techniques [1]. The following sections briefly review these techniques and introduce several typical related music recommender systems.

2.2.1.1 Collaborative Filtering

Collaborative filtering (CF) [131] is the most common approach in recommender systems. This technique relies on users' past behaviors and recommends items to a particular user if they are liked by similar users. A merit of this approach is that it does not need to analyze item contents. This is an important advantage in music recommendation, given the complexity of analyzing music contents. CF is to predict the relevance of items to a user based on the records of user ratings or implicit feedbacks. It has two categories of methods - memory based and model based [125].

Memory Based Methods are also called neighborhood methods in literature [73]. Memory based methods operate over the entire database to compute the relationship between items or users. Given a matrix R of dimensions

$|U| \times |I|$ to represent the rating data of all users on all items in a dataset. Each element $r_{u,i}$ in a row u denotes the rating of user u gave to item i. The rating could be binary (i.e., $\{0, 1\}$) in *implicit feedback*, or a real value in $[1, 5]$ in *explicit feedback* (e.g., the ratings in Netflix data [11]). An unknown rating of user u for item i can be predicted either by finding a set of users similar to u (user-based CF), or a set of items similar to i (item-based CF). Here we give the basic formulas for user-based CF. Given a user u and an item i, the predicted rating of this user to this item is:

$$\hat{r_{ui}} = r_u + K \sum_{v=1}^{n} w(u, v)(r_{vi} - r_v) \tag{2.1}$$

where r_u is the average rating of user u, n is the number of users in the database with known ratings for items i, $w(u, v)$ is the similarity of users u and v, K is a normalization factor to keep the sum of $w(u, v)$ is 1 [20]. Different methods have been proposed to compute the user similarity w [36]. Person correlation (2.2) [122] and Cosine distance (2.3) [130] are two most common measures:

$$w(u, v) = \frac{\sum_{j=1}^{k}(r_{uj} - r_u)(r_{vj} - r_v)}{\sqrt{\sum_{j=1}^{k}(r_{uj} - r_u)^2 \sum_{j=1}^{k}(r_{vj} - r_v)^2}} \tag{2.2}$$

$$w(u, v) = \frac{\sum_{j=1}^{k} r_{uj} r_{vj}}{\sqrt{\sum_{j=1}^{k} r_{uj}^2 \sum_{j=1}^{k} r_{vj}^2}} \tag{2.3}$$

where k is the number of items both users u and v have rated.

Model Based Methods try to explain the ratings by characterizing both users and items on latent factors discovered by latent factor models, such as Probabilistic Latent Semantic Indexing (PLSI) [53], Latent Dirichlet Allocation (LDA) [15] and Matrix Factorization [73]. PLSI and LDA will be introduced in Section 2.3. Next we will introduce MF methods, which is very popular in the recommender systems and has achieved the best performance in Netflix Prize [73, 11]. Model based methods map users and items into a latent

factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each user u is associated with a vector $p_u \in \mathbb{R}^f$, which measures to what extent the user u has interest in the f factors. Similarly, each item i is associated with a vector $q_i \in \mathbb{R}^f$, which shows to what extent each item possesses those factors. The dot product of the user's and item's vectors $q_i^T p_u$ characterizes the user's overall interests on the item's characteristics. The dot product is used to predict the rating of user u for item i:

$$\hat{r_{ui}} = q_i^T p_u \tag{2.4}$$

To learn the factor vectors q_i and p_u, the system minimizes the regularized squared error on the set of known ratings:

$$\min_{q*,p*} \sum_{u,i} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{2.5}$$

The second item in the equation is the regularization item to avoid overfitting. Model-based methods use pre-computed models to make predictions. An advantage of matrix factorization model to collaborative filtering is its flexibility in various data aspects, such as adding bias and temporal dynamics [73, 72].

Examples of CF-based Music Recommender System. An typical example of collaborative systems is the one used in Last.fm. Last.fm keeps users' listening behavior and calculates the distance between users. Recommendations are made for users with similar preferences. The success of CF method in Last.fm relies on the large amount of users and their playback records. In [139], a user-based CF method was used to estimate the similarity between users according to the rated songs by them. Then the songs of similar users are recommended to the targeted users. In [36], an item-based CF method was used to determine the similar artists to keep the broadcasted playlist coherent.

Advantages and Limitations. Collaborative methods measure the sim-

ilarity between different music tracks or different users based on the played history of tracks or users, while the relationship between music tracks and users are not captured. In other words, the intrinsic music preference of users on music cannot obtain. This is the underlying reason of the limitations of CF methods. CF methods are known to suffer from the cold start, data sparsity and long tail (or popularity bias) problems [126].

2.2.1.2 Content-based Approach

Content-based (CB) systems recommend items similar to the ones known to be liked by the users. Items are represented by feature vectors. In music recommendation, the feature vectors can be audio features (such as timbre and pitch), textual descriptions (e.g., tags) or metadata (e.g., artists). The key step of CB systems is to capture and represent user's preferences with the same features. Then given a new item, the CB systems could estimate whether the user would like the item based on user's feature vector and the item's feature vector.

Examples of CB-based Music Recommender System: Because music content is difficult to describe and represent, CB music recommender systems are considerably less than CF music recommender systems. An example of commercial content-based music recommender system is Pandora, which estimate the artist similarity and track similarity based on experts' annotations. There are also other content-based recommender systems in literatures, such as MusicSurfer [26] and Musiper [143]. MusicSurfer [26] is a content-based system for navigating large music collection. The system uses perceptual and musical audio signal features (e.g., rhythm, tonal, and key note) to estimate music similarity. Musiper [143] constructs music similarity perception models of its users by associating different music similarity measures to different users. The content-based method avoids the popularity bias problem and the problem of cold-start for new tracks. Due to the current limitation of automatic music

content description techniques, however, content-based methods are typically less successful than the collaborative filtering method [27, 74]. Besides, the songs recommended by content-based methods are lack of serendipity, as the recommended ones are the most similar tracks with respect to the ones that users liked. For example, content-based methods are less likely to recommend the musical style that the user never heard before.

Advantages and Limitations: The major limitations of content-based recommendation methods are inherited from the content-based music retrieval techniques - lack of highly effective scheme to compute music content descriptors about high-level concepts. Further, how to model user's music preference is also a major problem in CB music recommender systems. Content similarity cannot completely capture the preferences of a user, because of the semantic gap between the user's perception of music and the system's music representation. The third problem is that the recommended tracks may lack novelty, because content-based systems tend to recommend items too similar to those are used to define the user's profile. On the other hand, content-based systems can overcome the popularity bias in CF systems and the cold start problem of new tracks. However, new users are still an issue.

2.2.1.3 Hybrid-based Approach

As mentioned in previous subsections, collaborative methods and content-based methods suffer from their own limitations, respectively. Hybrid methods combine advantages of two methods and avoid their limitations based on the complementary nature of two methods. Burke [22] summarizes six methods of combining different recommendation techniques in hybrid recommender systems: weighted, switching, mixed, feature combination, cascade, feature argumentation, and meta-level. For details of each method, please refer to the review [22]. Only few music recommender systems have been reported in literatures [37, 165]. In [37], Donaldson et al. presented a hybrid recommender

system, which uses a feature combination method to combine item-based CF data with acoustic features. Yoshii et al. [165] used the three-way aspect model to associate ratings and content features with a set of latent variables.

2.2.2 Context-aware Music Recommendation

The recommender systems described above captures the long-term music preference of users. However, people often prefer different music under different contextual situations. Many external factors influence users music preference, including environmental-related context (e.g. location, time, temperature, ambient lighting conditions and background noise), user-related context (e.g. activity and emotional state), as well as social influence (e.g. friends' music preferences and the music popular trends). For example, when reading books in library, peaceful music is a good choice; while energetic music is preferred when running in gym. In [18], Lee et al. find that there is a growing need for contextual-aware music recommendation to provide better results. In recent years, more and more attentions have been devoted to the development of CAMR systems.

Table 2.1 presents a comprehensive overview on existing studies on context-aware music recommendation, from the aspects of considered contextual factors, used data in recommendation, recommendation methods/models, implementation and evaluation. From the table, we can see that a wide range of contextual factors have been studied and considered in CAMR systems, including environment-related (e.g., time [2, 4, 6, 17, 60, 65, 83, 82, 84, 112, 115, 119, 121, 123, 38, 138], location [2, 3, 17, 61, 84, 19, 98, 119], weather [5, 6, 34, 67, 83, 82, 112], noise [44, 112], temperature [34, 44, 67, 83, 82, 112], lighting [34, 44], etc.) and user demographic-related (e.g., age and gender [60, 67, 83, 82, 112, 123, 138]) and state-related contexts (e.g., mood [5, 6, 60, 65, 75, 123, 138], activity [6, 34, 44, 65, 138, 159], walking pace [40, 44, 163], heart

Table 2.1: An overview of context-aware music recommender systems.

CMAR Systems	**Contextual Factors**	**Used data for recommendation**	**Recommendation Method/Model**	**Prototype**	**Evaluation**
Foxtrot [3]	Location (geographic coordinates)	User scores for music in a location	Weighted combination of scores for a location	Desktop prototype	Simulation with 100 participants on desktop and evaluate their engagement
Incarmusic [5]	Driving style, road type, landscape, sleepiness, traffic conditions, mood, weather, and natural phenomena	Ratings	Matrix factorization	Android prototype	User study with 66 participants
Andromedia [17]	GPS, time, heart rate, time, and and local device information	Explicit feedback	Context reduction and context weighting	A prototype consisting of a central server and a PDA client	No evaluation
Adapted to POIs [18, 19, 61, 62, 63]	Points of Interests	User emotional annotations, music contents (for auto-annotation)	Jaccard similarity	Prototype on Android mobiles	User study with tens of participants
Saturday night or fever [34]	Activity, temperature, lighting, and weather	Emotion tags for music and contexts	Euclidean distance	No	Evaluated on 8 songs
Xpod [38]	Acceleration, skin temperature,heat flow, heat flow cover, time, transversal cadence, longitudinal cadence, day of week	Music metadata and beats	Machine learning algorithms, such as SVM, KNN, etc.	No	Evaluated on 239 songs
PersonalSoundTrack [40]	Walking pace	User logs and annotations of songs	Linear regression	A prototype on a laptop with accelerator connected via Bluetooth	No results given
Sonic city [44]	Light, noise, pollution level, temperature, electromagnetic, activity, enclosure, slope, presence of metal, heart rate, pace, and ascension/desent	Music content	Rule-based method (mapping context to music concepts, e.g., tempo, rhythm)	A wearable prototype	User study with 5 users

Music for my mood [82, 83]	season, month, weekday, weather, temperature, gender, age, and region	User listening logs, music genre	Case-based reasoning	Desktop prototype	Evaluation on 660 users
Supermusic [84]	Location and time	Listening history	Collaborative filtering	Symbian prototype	A five-week user trial with 42 users
Geoshuffle [98]	Location points in user's daily routines	Music metadata and audio features	Case-based reasoning	Phone/iPod prototype	User study on 14 users and evaluate the performance based on user's skip behaviors
Musicheart [104]	EGG, heart beat, and pulse	Music content, user listening logs	Estimate the rating by considering the similarities among songs and other suer's rating	Android prototype	User data collected from 37 participants for heart rate detection and 17 participants for activity level inference. Performance is evaluated on 4 users.
CAMR by fuzzy Baysian networks [112]	Temperature, noise, humidity, illuminance, weather, gender, age, season, time	User rating and music annotations	Fuzzy Bayesian networks with utility theory	Desktop prototype	Desktop simulation performed with 10 participants on 322 music pieces
Lifetrak [119]	Location, time, kinetic, entropic, and meteorological	User rating	A heuristic combination method	A prototype on Nokia 770	No evaluation
Time-aware [121]	Time (day, week)	User listening history	Circular statistical analysis	No	Evaluated by predicting user's music selection on artist level and genre level
CAMR for daily activities [159]	Activity	Music content	Bayesian probability model	A prototype on server and Android phones	User study with 10 participants
Painting the city [2]	Location, time	Music metadata	Case-based reasoning	No	No
Personalized music system [163]	Pace	Music metadata and tempo	Matching tempo (bpm) with stride frequency (spm)	Desktop prototype	User study with 6 participants
Time-dependent [4]	Time (hour, day, month, year)	Implicit user feedback	Factorized-based CF	No	Evaluated on 338 Spanish user data from Last.fm
Music mood classification [123, 60]	Age, sex, job, hobby, mood, time, location, and event	Rating and music content	Collaborative filtering and ontology-based reasoning	Desktop prototype	User study with 30 users and 120 songs
WhozThat [10]	Local proximity context (e.g., bar)	User listening logs	Collaborative filtering	A prototype on Nokia N80	No

A mobile-based system [65]	Location, activity, time, and venue	Metadata, user profiles	Case-based reasoning	No	A small test lasting one week with 10 users, and then 59 users in 4 different countries
Emotion-based system [75]	Emotion	Film music, music contents	A modified affinity graph	No	Evaluations on 107 file music pieces from 20 animated films
Design of MRS [67]	Sex, age, pulsation, weather, temperature, and location	Music metadata, user listening logs under different contexts	A statistical filtering method	Desktop prototype	50 users participated in system evaluation with 100 songs
Interactive system [138]	Mood, identity, location, activity, time, and demographics	Listening logs under contexts, music metadata	Context-aware collaborative filtering	A prototype on Nokia N95	No evaluation
Playlist recommendation based heartbeat [90]	Heartbeat	Music medatada, tempo	Markove decision process	No	User study with 6 participants
Usage context prediction [6]	Activity, weather, time of day, and mood	Rating	Collaborative filtering	No	Evaluation based on 74 users labeled data on 100 music tracks along 14 contextual conditions
Music for seasons [115]	Seasons	Music genre	No recommendation model	No	User study on 232 male and 199 female college students
Emotion-based slideshow [88]	Image	Music content, image content	Associate painting and music based on emotions	No	Subjective evaluation are performed with 18 participants on 138 impressionism painting and 160 classical piano compositions
Picasso [145]	Image	Music content, image content	K-nearest neighbors with smoothing techniques	No	Evaluation performed with 13 participants
Emotion context from article [30]	Articles	Music content, text documents	Factorization machine-based method	No	Evaluated on a dataset collected from LiveJournal
Musicsense [24]	Text of the viewed web page	Music metadata and reviews	A generative model for mood allocation	No	Evaluation preformed with 100 songs and 50 web logs
Just-for-Me [31]	Venue and popularity trends	Music content, user listening logs	A unified probabilistic generative topic model	Prototype on Window Phone 8	Evaluation performed by offline and online experiments with more than 1000 songs
VenueMusic [33]	Venue	Music content, user listening logs	A unified probabilistic generative topic model	Android prototype	Evaluation performed by offline experiments on more than 1000 songs and online experiments on 10K songs

beat rate [17, 44, 90, 104, 38], etc.). However, the development of CAMR is still in the early stage and very few CAMR systems can be applied in the real world, due to the challenges on the following aspects:

- Context detection - Various types of contexts could affect users' music preferences. Among them, some contexts could be detected at high accuracy by mobile devices (used as music players), such as time, location, and weather, while other contexts are hard to detect by most of the current mobile devices, such as heart rate and mood. Many of the existing systems require the installation of special sensors in the system or wearable sensors in human body to detect the contexts for music recommendation [17, 40, 44, 90, 104, 38]. These systems are not feasible in real applications.

- Dataset collection - Most of the CAMR systems rely on users' listening logs or ratings to train the recommendation models. However, it is difficult to collect such data, especially for the systems which consider many different contexts, such as [5, 34, 17, 44, 60, 83, 82, 112, 119, 123, 38, 67, 163], because it requires extensive user's labor efforts to label different contexts associated with the listening logs or ratings.

- System evaluation - How to evaluate the performance of developed CAMR systems is also a very challenging problem. It can be observed that many systems have not been evaluated at all in the reported literature, such as [2, 17, 40, 119, 138]. Because the complexity of different contexts and the difficulty in collecting data, there is no standard test collections for the evaluation of CAMR systems. Accordingly, it is hard to fairly and comprehensively evaluate the performance of the developed systems. Besides, there is no standard methodology for CAMR system evaluation. In existing studies, researchers construct a small collection with hundreds of songs by recruiting a small group of users for labeling/rating the dataset

and evaluating the performance of the systems [3, 4, 6, 5, 34, 44, 60, 65, 67, 75, 84, 90, 98, 104, 112, 24, 123, 121, 38, 159, 163]

- Context-aware music preference modeling - Obviously, context-aware music recommendation modeling is more difficult than traditional music recommendation modeling, which has not considered the factor of context. Collaborative filtering is the most popular and success model in traditional recommendation modeling, which is also used in CAMR (e.g., [2, 4, 5, 84, 138]). However, a major problem in CF is sparsity, which becomes much severer in CAMR, since it is much harder to collect records under each type of contexts. As a result, in most systems, heuristic recommendation methods are used, such as nearest neighbors [145], simple similarity matching [34, 61, 62, 104], and case-based reasoning [44, 65, 83, 82, 98]. Music content has not been well explored in the CAMR model. Many systems only used the listening logs or ratings with the associated contexts for recommendation without the analysis of music contents, which easily suffers from the problem of the "cold start" for new music items. In other cases, content is either used to annotate the music items with context labels (e.g., moods [88, 75]) or used to measure the similarity between two songs [84, 146]. None of the previous systems analyzes the music preferences of users under different contexts based on the music content.

2.2.2.1 Location-aware Music Recommendation

Location has a strong impact on user's music preferences [19]. North et al. [105] presented a study to explore five aspects of the ways how people use music in everyday life. They found that in different places, people listen to music for different reasons. For example, when people in the office, they listen to music because the music helps them concentrate/think; while in pub/night club, the

music is used to create the right atmosphere. The observations imply that the place where to listen to music plays an important role in users' local music preferences. Although the importance of location on users' music preference, there is little research exploring the location-related information in music recommendations. In [44], Gaye et al. designed a prototype of an interactive music system to generate electronic music for urban environments. The system heavily relies on hardware to collect various user-related (e.g., heart rate, arm motion) and environment-related contextual information (e.g., light, temperature, and noise, etc.). Lifetrak [119] considers the location (represented by a ZIP code), time, weather and activities to generate a playlist based on user's music library. A mobile audio application Foxtrot [3] allows users to assign audio content to a specific geo-location, and play audio content associated with a particular location. Kaminskas et al. [19, 63] conducted a series of studies on recommending music to the place of interests (POIs). They match the POIs and music by exploiting semantic relations between the POIs and music items with the assigned emotional tags to both POIs and songs. Most of these studies relate the location information with geographical coordinates. However, it is hard to capture the correlations between music contents and a specific geo-location. As a result, for a location, these systems can only recommend the songs liked by users in this location based on previous records [119, 3]. It is worth to mention that the POIs in Kaminskas et al. [19, 63] are the places where people do not visit frequently in everyday life.

Going beyond the geo-location information of latitude and longitude, each venue possesses its own distinguishing atmospheres or semantics. GeoShuffle [98] considers the effects of the locations at where users usually listen to music in their daily lives. The key difference is that in GeoShuffle, the location was captured based on GPS data and the locations considered are restricted to the points in people's daily routines. Listening records are used to capture a user's music listening habits while in the routine paths. Therefore, its

performance depends on both the regularity of user's daily routines and the quality of historical preference data. In our previous work [31], a Just-for-Me music recommender system was developed for effective personalized music recommendation in different types of venues, together with the consideration of global music popularity trends. Just-for-Me applies an extended three-way aspect model and represents each song as a "bag-of-audio-words" document to learn the topics. In the extended three-way aspect model, users' music interests are represented as topic distributions, and topics are the distributions of songs, venues, and audio words. Inspired by the key research findings about the strong influence of venue type on users' music preference, we focus on the problem of recommending suitable songs based on different types of venues in Chapter 3. Core innovation of our proposed VenueMusic system in this chapter is a location-aware topic model (LTM), which naturally associates venue types and music contents in a latent semantic space by using "bag-of-words" based representation.

2.3 Topic Model

Topic models, such as probabilistic Latent Semantic Indexing (PLSI, also called aspect model) [53] and Latent Dirichlet Allocation (LDA) [15], are originally proposed to discover the underlying *themes* or *latent topics* of a large scale of text documents. The latent topics are discovered by mining co-occurrence patterns of words in documents that exhibit similar patterns. Base on the topic models, each document is represented as a multinomial distribution over the latent topics, which are in turn multinomial distributions of terms. The basic idea of topic model is that there are latent topics to explain the occurrences of words in the documents of a corpus. Each document has its own topic distribution, and each word in a document is associated with a latent topic. Thus, the corpus can be summarized by the latent topics, and each document

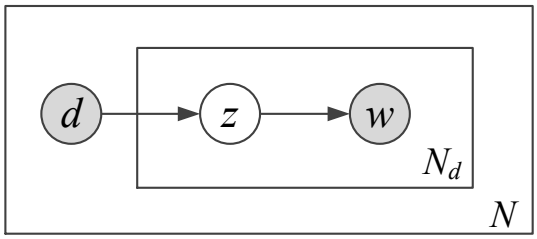


Figure 2.1: Plate notation of PLSI model.

is represented by the probabilistic distribution of the latent topics.

2.3.1 Probabilistic Latent Semantic Indexing

The PLSI is proposed by Hoffman [53]. The graphic model of PLSI is shown in Fig. 2.1, which describes the generative process of each of the N documents in the collection. In the figure, N_d denotes the number of words in document d. Each word w has associated a latent topic z, from which it is generated. The shaded circles indicate observed variables, while the unshaded one represents the latent variables. The graphic model of PLSI is shown in Fig. 2.1. The generation process of a word w in a document d can be expressed as

$$P(w|d) = P(w|z)P(z|d) \tag{2.6}$$

The equation describes that the authors select a topic z according to the topic distribution $p(z|d)$ of the document d, and then select the word w based on the word distribution given the topic z. Repeating the generation process in sufficient times, we can finally generate a full document and eventually the whole document corpus. PLSI shows a sound probabilistic generation model, however, it is poor on prediction unobserved words and documents. According to the equation, the joint probability of generating the words and documents

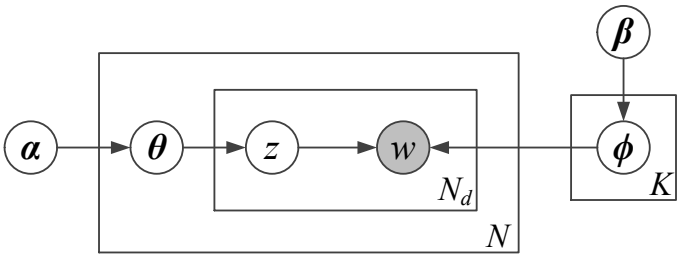


Figure 2.2: Plate notation of LDA model.

in the corpus can be expressed as,

$$\begin{aligned} P(d,w) &= \prod_{i=1}^{N}\prod_{j=1}^{N_d} P(d_i,w_j) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{N_d}\sum_{k=1}^{K} P(w_j|z_k)P(z_k)P(d_i|z_k) \end{aligned} \tag{2.7}$$

where K is the number of latent topics. The model parameters are estimated by maximizing the log-likelihood using Expectation-Maximization algorithm [35].

By treating *users* as *documents* and *items* as *words*, Hofmann and Puzicha [54] applied the aspect model to user-item co-occurrence data for collaborative filtering. Latent topics are discovered based on the item co-occurrences across different users' profiles; then a user's interest is represented as a distribution of the latent topics. Later on, Popescul et al. [116] extended this method to incorporate the item's content for discovering user's interests based on both of the co-occurrence of items among users' profiles and the co-occurrence of item contents in items. The method is also called three-way aspect model [116].

2.3.2 Latent Dirichlet Allocation

The graphic model representation of LDA model is shown in Fig. 2.2. The shadow node denotes observed parameters; the transparent nodes are the hidden parameters. The basic idea of LDA is similar to PLSI, while LDA introduces two Dirichlet vector priors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, as shown in Fig. 2.2. The function of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is to constraint $p(z|d)$ and $p(w|z)$, respectively, so as to solve

the "overfitting" problem in PLSI. Let $\boldsymbol{Z} = \{z_1, z_2, ..., z_K\}$ denote the topic vector, and K is the number of topics. $\boldsymbol{\alpha}$ is a K-dimensional vector. Each element $\alpha_i \in \boldsymbol{\alpha}$ is a prior for a corresponding element $z_i \in \boldsymbol{Z}$. A higher value of α_i will increase the probability of observing topic z_i in the corpus. Similarly, let $|W|$ denote the number of distinct words in the corpus, $\boldsymbol{\beta}$ is a $|W|$-dimensional vector. Each element $\beta_i \in \boldsymbol{\beta}$ is a prior for a corresponding word $w_i \in \boldsymbol{W}$. A higher value of β_i will increase the probability of $p(w_i|z)$. Thus, the Dirichlet parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ smooth the multinational distributions of $p(z|d)$ and $p(w|z)$. Assigning smaller values to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ will reduce the smoothing effect and result in more decisive topic associations [49]. In other words, the value of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ control the sparsity of the document-topic distribution and topic-word distribution, respectively.

Various approximation inference methods have been developed to estimate the parameters in variants of LDA, such as variation inference [15], expectation propagation [99], and collapsed Gibbs sampling [46]. Although Gibbs sampling is not necessarily as computationally efficient as approximation schemes such as variation inference and expectation propagation, it is unbiased and has been successfully applied in many large scale applications of topic models [46, 128, 149, 103]. In this thesis, we apply collapsed Gibbs sampling to estimate the parameters in the proposed topic modes in Chapter 3 to Chapter 5.

There are many variations of topic models. In the following, we briefly review hierarchical topic models and multimodal topic models, which are related to our proposed models in this thesis.

2.3.2.1 Hierarchical Topic Model

Hierarchical topic models are able to obtain the relations between topics, such as nested Chinese Restaurant Process (nCRP) [13], tree-informed LDA [66] and nHDP [110]. nCRP [13] uses Chinese restaurant process as a representation of prior and posterior distributions for learning hierarchical topics. The learned

topics are organized in a tree structure, in which each topic is a node in the tree. The topics of a document are restricted to follow a path from root to leaf in the tree. nHDP [110] generalizes the nCRP framework to allow that the topics of a document could access the entire tree. Tree-informed LDA [66] also learns the tree-structure topics and it parameterizes how closely the topic proportion of a parent topic are inherited by its children topics. Many supervised hierarchical topic models have also been proposed to include the structure information of labels, such as HLLDA [114], HSLDA [113], and SSHLDA [95]. A common characteristic of these hierarchical topic models is that they all focus on modeling the parent-child and sliding relations between topics. Besides, the topics in those models are represented as the mixture of words. Thus, these topics (no matter parent topics or child topics) are all in the same semantic space. Distinguished from these models, the Dual-Layer Music Preference Topic Model (DL-MPTM) in Chapter 5 discovers two sets of latent topics under dual-layer latent spaces: the latent topics in the high-layer latent space are the mixtures of the latent topics in the low-layer latent space.

2.3.2.2 Multimodal Topic Model

Since the success of LDA in single modality scenarios, it has also been extended to multi-modal cases. Some multi-modal topic models have been proposed in literature, such as mmLDA [7], Corr-LDA [14], tr-mmLDA [117], MDRF [59], and factorized multi-modal topic model [155]. The basic philosophy behind these multi-modal LDA is the existence of shared latent topics that are the common causes of the correlations between different modality. In mmLDA [7], the image and text words are generated from two non-overlapping sets of hidden topics. For an image, the two sets of topics follow the same topic distribution. Corr-LDA [14] is designed so that image is the primary modality and is generated first, and each caption word is forced to be associated with an image region and is generated based on the topic of this image region.

Tr-mmLDA [117] uses a latent variable regression approach to learn a linear mapping between the topic distribution between two modalities. Factorized multi-modal topic model [155] generalizes the modeling of two modalities to multiple modalities. It models the dependencies between topics both within and across modalities by introducing auxiliary variables. These models assume there exists correspondences between different modalities, such as a corresponding text document for an image document. Multi-modal document random field (MDRF) [59] model learns a topic model from a set of documents by using a document-level similarity graph, which models the similarity between different documents.

The DL-MPTM in Chapter 5 uses Corr-LDA in the dual-layer structure, as there are only two modalities - audio and text - in our problem. Besides, because social tags are usually incomplete, the text document (formed for a song) is not complete as a corresponding document to the audio document of the song. The merit of Corr-LDA is that the topics of text words are indeed a subset of topics that occur in the corresponding image (song in our context), and an audio segment could be associated with multiple text words, which is reasonable for the annotation of textual concepts to an audio segment.

Chapter 3

Venue-aware Music Recommendation

In this chapter, we present a novel venue-aware music recommender system called VenueMusic to effectively identify suitable songs for various types of popular venues in our daily lives. Towards this goal, a Location-aware Topic Model (LTM) is proposed to 1) mine the common features of songs that are suitable for a venue type in a latent semantic space and 2) represent songs and venue types in the shared latent space, in which songs and venue types can be directly matched. It is worth mentioning that to discover meaningful latent topics with the LTM, a Music Concept Sequence Generation (MCSG) scheme is designed to extract effective semantic representations for songs. An extensive experimental study based on two large music test collections demonstrates the effectiveness of the proposed topic model and MCSG scheme. The comparisons with state-of-the-art music recommender systems demonstrate the superior performance of VenueMusic system on recommendation accuracy by associating venue and music contents using a latent semantic space.

3.1 Introduction

Intelligent recommender system, as a promising technology for music search, aims to assist users in exploring large scale music collections by identifying *suitable* songs based on their preferences. Users generally prefer music players which can automatically recommend the playlists fitting their preferences based on current contexts (e.g., mood, location, event and activity). Indeed, a wide range of contextual information have been recently explored in the music recommender system development [126]. These contexts include both environment-related (e.g. location and time) [19, 5, 31, 132, 135] and user-related contexts (e.g. activity and emotion) [24, 159]. These studies have demonstrated that the incorporation of contexts in recommendation can effectively enhance the user's satisfaction on recommendation results. As a matter of fact, location is one of the most crucial contexts and has significant influence on user's music preference [19, 106]. Several previous studies attempted to recommend music to specific geo-locations [119, 3]. Besides, Baltrunas et al. [5] built an in-car music player for recommending music to the landscapes passed when driving a car. Kaminskas et al. conducted a series of studies on retrieving songs suited for place of interests (POI) based on emotional tags [19, 62, 63]. However, one important context that is generally ignored in current research is user's venue. To the best of our knowledge, no existing approaches can effectively recommend music based on common venues, such as *office*, *library*, *gym*, *mall*, etc.

Venue, referring to *the place where activity or event happens*, is an important location based context and becomes more and more important in music recommender system design and development. On the one hand, different types of venues are where people usually listen to music in everyday life [106]. On the other hand, every day people could enjoy music at different types of venues, where different surrounding environment and atmosphere can be found. Thus,

venue type has important influence on users' song selections and suitable songs can be very helpful to create the nice atmosphere for a particular venue. For example, *night bar*, *restaurants* and *shops* often use music to help them create the right atmosphere for their customers. Furthermore, users' activities, which also play a critical role in determining users' song preferences [159, 106, 85], highly correlate with venue type. In fact, when users are engaging in the same or similar activity, the songs they prefer or play share many common musical characteristics [159]. For example, low tempo and middle-pitch-range music is usually selected to assist users in concentrating or thinking, while up-tempo music is a nature choice for physical exercise in the gym and dance party in Disco.

This study mainly focuses on the effects of venue types instead of geo-locations (a geo-location refers to a point pinpointed by geographic coordinate), because users' music preferences are more likely to be influenced by the atmosphere and environment of venue types. For examples, a user would prefer similar types of music when he is working out no matter in the gyms nearby his office or the ones nearby his home, although these gyms have different geo-locations. In addition, when conducting different activities in a venue, it is often that users might like the same type of music, such as when reading and writing in library. To support efficient music access, listeners frequently organize songs into different playlists, which are suitable for various venue types. For example, in a popular music streaming service website Grooveshark[1], venue types are very common titles of their playlists. It is often that the same song appears in many different playlists named with the same venue type but created by different users (refer to Sect. 3.3.1.2). This observation suggests that users share similar understanding and view about the music contents suitable for a particular venue type. For the simplification of presentation, unless otherwise indicated, *venue* in this chapter refers to *venue type* hereafter.

[1]http://grooveshark.com

Motivated by the earlier discussions, we study the problem of recommending suitable songs to different types of venues by exploring the correlation between the music *features* and the *characteristics* of these venues. In general, a venue owns its distinct *characteristics*, such as ambience and atmosphere. Songs with certain *features* that fit those *characteristics* could be more suitable for this particular venue, such as energetic music for gym and peaceful music for library. According to the study [77], users tend to label the pieces of music they like using high-level concepts, such as styles and emotions. It reveals that human perceives and judges music based on the semantics embedded in music contents. In many cases, music semantic meaning cannot be explicitly described and characterized using low-level spectral features due to the well known "semantic gap" [169]. Acoustic contents belonging to same or similar concepts could be highly diverse. Furthermore, a song could include a complex mixture of concepts at different levels. Therefore, the utilization of acoustic features or concepts for describing music preferences at a venue may not be effective and comprehensive enough to support high quality recommendation.

In this chapter, we present a smart music recommender system called VenueMusic, which can automatically generate a playlist matching a target venue appropriately [32]. Towards this goal, we approach the problem from a new perspective of effective topic modeling and develop a novel scheme called Location-aware Topic Model (LTM), which models the associations between the music contents and venues in a *latent semantic space*. Similar to the standard Latent Dirichlet Allocation (LDA) [15], in the LTM, each topic is a multinomial distribution of music semantic concepts, which captures the interactions between various music semantics. Each venue and each song are then represented by the multinomial distributions of these latent topics. Intuitively, the topic distribution of a venue characterizes *relevant music properties* of songs that are suitable for this venue; and the topic distribution of a song reflects how general users perceive the music. As both songs and venues are

represented by the same latent topics, the suitability of a song for a venue can be directly measured. The LTM is trained based on a set of songs labeled with different venues. To enable the LTM to characterize the semantic meaning of a song, each song is represented as a "bag-of-words" document. This is different from the existing methods [165, 31] based on "bag-of-audio-words", which can not effectively express the semantic meanings of a song. In the VenueMusic system, each song is represented as a sequence of *music concepts*[2], i.e., a "bag-of-text-word" document. In particular, a Music Concept Sequence Generation (MCSG) method (Sect. 3.2.2) is proposed to generate the concept sequence of a song. As validated in our experiments, song representation based on semantic concept sequences in the LTM is more effective than those using low-level "audio words". Our main contributions can be summarized as follows:

- A location-aware music recommender system is developed to recommend music to different types of common venues in everyday life. The system matches songs and venues based on their semantic features. This is the first attempt on developing venue-aware music recommendation methods.
- A novel topic model LTM is proposed to capture the natural connections between the venue semantics and the music *contents*. The latent semantic topics extracted by the LTM are used to characterize the *music features preferred in different venue types* as well as *the music features of songs*. With this approach, the suitability of a song to a venue can be quantitatively measured in a latent semantic space.
- A *semantic concept sequence generation* scheme is designed to represent a song as a set of concepts for topic modeling. Besides, an *infrequent concept pattern filtering method* is introduced to remove noisy concepts in the generated semantic concept sequence. The final semantic concept sequences of songs are effective on the training of LTM.

[2]A music concept could be one or several text words that is usually used to describe music, such as *genre* and *mood* words.

- Two large scale music test collections are constructed to evaluate and compare the performance of our system with a set of competitors over a wide range of venues. The core empirical results demonstrate the potential of our VenueMusic system.

The rest of the chapter is organized as follows. The framework of the music recommender system is presented in Section 3.2. Section 3.2.3 introduces the LTM and provides details about algorithms for the model parameter inference. Section 3.3 describes the experimental configurations. The evaluation results are presented and analyzed in Section 3.4. Finally, Section 3.5 concludes the chapter with discussion of the findings in this study and directions for future research.

3.2 The VenueMusic System

3.2.1 System Overview

The VenueMusic system consists of two main functionality modules: Music Concept Sequence Generation (MCSG) and Location-aware Topic Model (LTM). Fig. 3.1 illustrates details of the system architecture. Given a set of songs labeled based on their suitabilities to venues (venue-labeled music collection), each song is represented as a Music Concept Sequence (MCS) via MCSG module. Then the LTM is trained to discover a set of latent topics, which form a latent space. Both songs and venues are represented as topic vectors in this latent space. For new songs in a music dataset, they are automatically converted into a MCS in the same way and mapped into the same latent space by the topic model. With the representations of songs and venues in the same space, the relevance (or suitability) of a song with respect to a venue can be directly measured. Since topic vectors are probabilistic distributions of the latent topics, the relevance between a song s and venue type l are evaluated

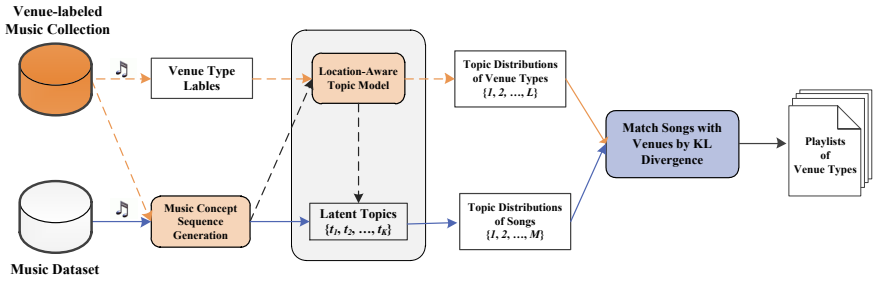


Figure 3.1: The framework of VenueMusic System.

using Kullback-Leibler (KL) distance. Specifically, a song s and a venue type l are both represented by the probabilistic distributions of K topics, the KL distance is expressed as:

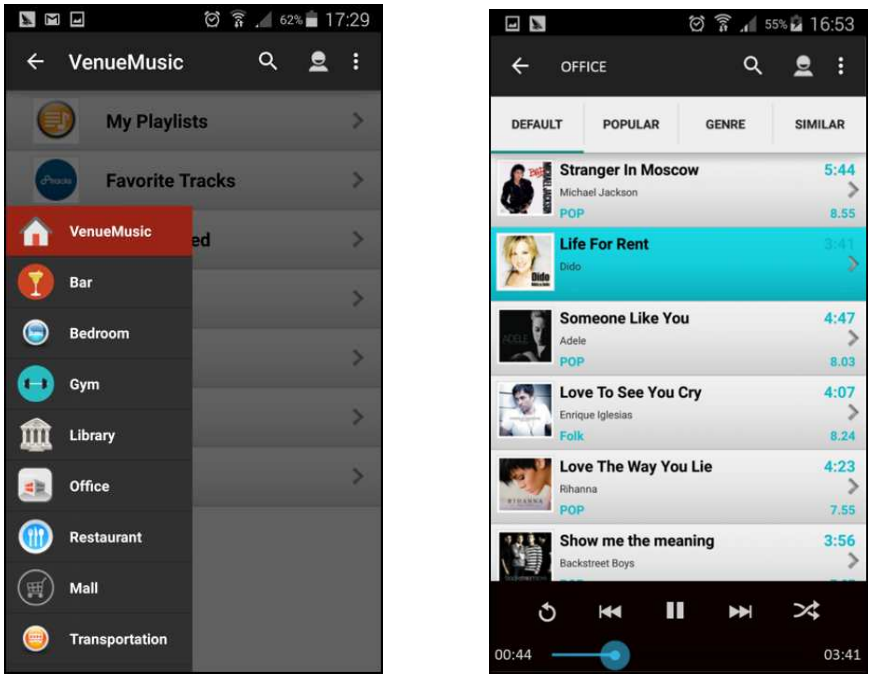
$$KL(l||s) = \sum_{k=1}^{K} l(k) ln(\frac{l(k)}{s(k)}) \tag{3.1}$$

where $l(k)$ and $s(k)$ are the probability of k-th topic in the topic distribution of l and s, respectively. The system is designed based on the key observation that a particular venue owns its distinct *characteristics* or *atmosphere*, which closely associates with the *events* or *activities* occurring in this venue. Typically, different types of music can be applied to match the atmosphere or activities in different venues [159, 62, 124]. VenueMusic aims to model those rich and complex associations effectively and comprehensively via the LTM.

A prototype of the system has been implemented on Android platform (Android version 4.4, 2GB RAM, Samsung Galaxy S5). Figure 3.2 shows the screenshot of the user interface of the system. The design of the interface is to facilitate the easy interaction between user and VenueMusic and enable users to smoothly access music services. For more details, please refer to [32].

3.2.2 Music Concept Sequence Generation

The most straightforward scheme to generate a sequence of "word units" about music contents is "bag-of-audio-words", which has been explored in many studies [165, 127, 56]. However, this approach suffers from a few limitations. First,



(a) (b)

Figure 3.2: The user interface of VenueMusic prototype

"audio words" are representative audio frames and thus have no semantic meanings. In the real world, people characterize music contents using music semantic concepts (e.g. *mood*, *genre*, *instrument*, etc.), which reflect how human perceives and interprets acoustic content. It is very difficult to connect the topics generated based on "audio words" with these music concepts. Second, the number of "audio words" is hard to be determined. A small number of "audio words" will not be able to represent and distinguish different music contents effectively, while a large number of "audio words" will lead to sparsity problem and low efficient indexing and learning.

To address the issues of "audio words", we develop a method to extract semantic music concepts (e.g., genre, mood and instrument) from the audio contents to represent a song as a MCS, which is the concatenation of concepts in small segments of the song's audio stream. Alternatively, we can represent each song by assigning music concepts to the whole song. Comparing with this alternative method, MCS has at least two advantages: (1) good com-

prehensiveness: it contains all the possible music concepts expressed by the audio contents; and (2) good differentiation: it can differentiate the relatively important concepts for a song. For example, in a song, the more segments a concept appears in, the more important or representative this concept is for the song. By aggregating a large set of songs for a venue, the latent associations between the music concepts for this venue can be mined from the MCSs of these songs. The quality of music concept sequence is very important for discovering such latent associations. To improve the concept detection quality, two post-filtering procedures are designed to reduce noisy concepts. As illustrated in Fig. 3.3, MCS generation consists of three main steps:

1. Partition a song into multiple segments;

2. Estimate the probability of each music concept in each segment using concept detectors based on the extracted audio features, and then filter the concepts to keep the most representative and confident concepts of the segment via two *filtering* methods;

3. Concatenate the remained concepts of each segment to form the MCS for this song.

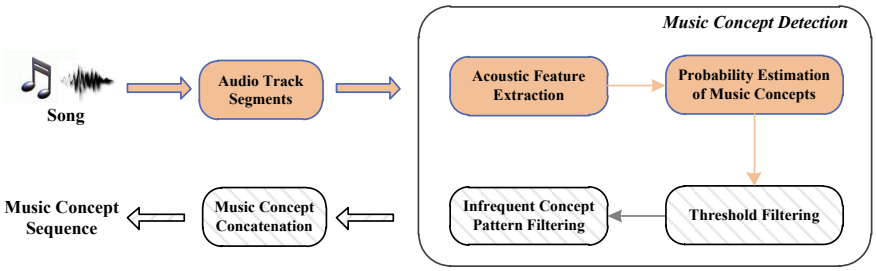


Figure 3.3: Architecture of semantic concept sequence generation.

The segments can be obtained by simply cutting the audio stream of a song into fixed-length windows or by detecting segments using music segmentation methods [92]. In our implementation, the former method is applied due to its simplicity. Since step (1) and (3) are straightforward, we focus on the

description of step (2). There are three key components in step (2): *Audio Feature Extraction*, *Music Concept Probabilistic Estimation* and *Concept Filtering*. Fig. 3.4 gives very comprehensive illustration about system architecture of Music Concept Probability Estimation and Concept Filtering.

3.2.2.1 Audio Feature Extraction

For each segment, we extract four types of acoustic features:

- **Timbral feature:** It characterizes the timbral properties of music sounds. Timbral feature is calculated based on the short time Fourier transform, including *Mel-Frequency Cepstral Coefficients* (MFCCs) [91], *Rolloff*, *Flux*, *Low-Energy feature* [154], and *Spectral Contrast* [93]. The total dimensionality is 23.

- **Spectral feature:** It describes the spectral properties of music signal. They include: *Spectral Centroid*, *Spectral Asymmetry*, *Kurtosis*, *Audio Spectrum Flatness*, *Spectral Crest Factors* [21], *Slope*, *Decrease*, *Variation*; *Frequency Derivative of Constant-Q Coefficients* [137]; *Octave Band Signal Intensities* [42]. The total dimensionality is 70.

- **Rhythmic feature:** It represents the patterns of a song over a certain duration. In this study, our rhythm feature includes *Beat Histogram*, *Rhythm Strength*, *Regularity and Average Tempo* [93]. The total dimensionality is 12.

- **Temporal feature:** It characterizes the musical properties based on time domain signals. It includes: *Zero Crossing Rate*; *Autocorrelation Coefficients* [42]; *Waveform Moments* [42]; *Amplitude Modulation* [42]. The total dimensionality is 62.

Three public toolboxes are used to extract all the above acoustic features: MIR Toolbox [79], Yaafe [96], and Essentia [16][3].

3.2.2.2 Music Concept Probability Estimation

Music concept probability estimation aims to estimate the probabilities of various music concepts for a music segment, as illustrated in part (a) of Fig. 3.4. Suppose there are n music dimensions $\{C_1, C_2, ..., C_n\}$ (e.g., *genre*, *mood* and *instrument*) and N_i concepts for each dimension C_i, the probabilistic vector of a dimension C_i is $C_i = \{P_{i1}, P_{i2}, ..., P_{iN_i}\}, (0 \leq P_{ij} \leq 1, 1 \leq j \leq N_i)$, where P_{ij} is the probability that the segment belongs to j-th concept of C_i. Many existing regression and classification methods can be used to estimate P_{ij}. In our implementation, the Support Vector Machine (SVM) method in LIBSVM library is adopted for the task [29].

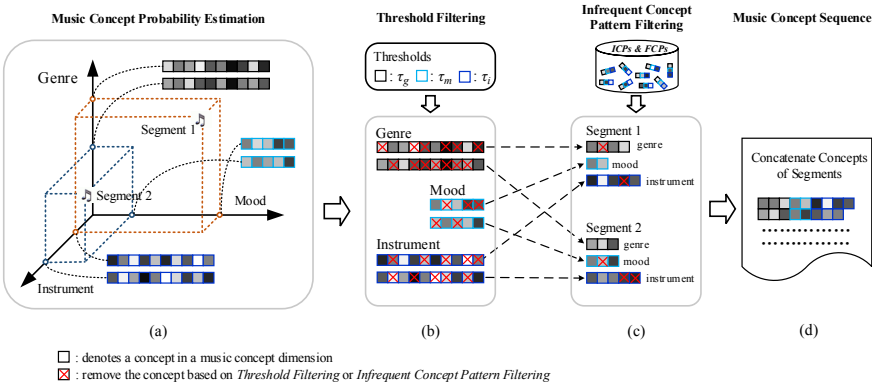


Figure 3.4: Illustration of the music concept probability estimation and concept filtering.

3.2.2.3 Concept Filtering

Generally, a music segment contains only limited amount of concepts in a music dimension. For example, it is really rare that a music is played with all kinds of

[3]Specifically, Yaafe was used to extract the following features: *Spectral Crest Factors*, *Slope*, *Decrease*, *Variation*, *Frequency Derivative of Constant-Q Coefficients*, *Octave Band Signal Intensities*, *Beat Histogram*, *Autocorrelation Coefficients*, *Waveform Moments*, and *Amplitude Modulation*; Essentia was used to extract *Spectral Contrast*; and other features were extracted by MIR Toolbox.

instruments. Thus, effective and comprehensive music characterization might not be achieved by using all the concepts. How to select the most representative concepts and remove noisy concepts becomes very important. In VenueMusic, two different strategies are proposed for the concept space refinement and their details are as below.

Threshold Filtering It aims at removing the concepts with a probability lower than a pre-defined threshold. Specifically, for each concept dimension C_i, there is a predefined threshold τ_i. If $P_{ij} < \tau_i$, then the j-th concept in C_i is removed, where P_{ij} indicates the probability of the j-th concept of C_i in a segment. The threshold filtering is illustrated in part (b) in Fig. 3.4. This filtering process is conducted in each music dimension separately.

Infrequent Concept Pattern Filtering As existing music concept classification algorithms cannot obtain very accurate results [39], it is possible that there are still mis-classified concepts remained after threshold filtering. To further improve the quality of generated concept sequences for songs, we propose an *Infrequent Concept Pattern Filtering* (ICPF) method. The underlying assumption is that there exist inherent interactions between concepts in different music dimensions, such as the use of *instruments* in different *genres*, and the expressed *moods* of certain *instruments* and *genres*. Although a piece of music can contain or express any combination of concepts, some of them are

Table 3.1: Few examples of Frequent Concept Patterns and Infrequent Concept Patterns discovered in our dataset. Each concept pattern is comprised by a concept from each of the three music concept types: *mood*, *instrument*, and *genre*.

Frequent Concept Patterns	**Infrequent Concept Patterns**
aggressive, guitar, rock	literate, snare, hiphop
literate, saxophone, country	humorous, clarinet, funk
rollicking, guitar, electronic	rollicking, snare, hiphop
passionate, violin, electronic	aggressive, clarinet, funk
aggressive, drumkit, alternative	humorous, drumkit, classical

very rare. For example, *guitar* is a popular instrument to express *passionate* mood in *rock* music, while *drumkit* has less chance to be found in *classical* music to express *humorous*. A *concept pattern* is comprised by a concept from each of the music dimensions. For example, suppose there are three music dimensions: *mood*, *instrument* and *genre*, then {*passionate*, *guitar*, *rock*} is a concept pattern. Infrequent Concept Pattern (ICP) indicates the concept patterns which are rarely found or even do not exist in a large music corpus, such as {*humorous*, *drumkit*, *classical*}. The music dimensions and corresponding concepts used in this study are discussed in Sect. 3.3.1.1 and shown in Table 3.2. Table 3.1 shows some examples of Frequent Concept Patterns (FCPs) and ICPs. The ICPF process is to remove the suspicious concepts that cause such rare combinations. The intuition is that the appearance of ICP is due to the mis-detected concepts. Detail steps of the ICPF process are as follows:

- Step 1 - Concept Pattern Construction: For a segment of a song in the dataset, after concept probability estimation and threshold filtering, a set of concepts of different music dimensions are obtained. With the obtained concepts, all the concept patterns of this segment are formed based on the concept pattern definition. For example, suppose three music dimensions are considered and for a segment, the obtained concepts are: three concepts in the first music dimension $\{c_{11}, c_{13}, c_{15}\} \in C_1$, two concepts in the second music dimension $\{c_{22}, c_{24}\} \in C_2$, and two concepts in the third music dimension $\{c_{32}, c_{37}\} \in C_3$. Then in this segment, 12 concept patterns can be formed, such as $\{c_{11}, c_{24}, c_{37} | c_{11} \in C_1, c_{24} \in C_2, c_{37} \in C_3\}$.

- Step 2 - FCP Set and ICP Set Construction: Count the frequency of each concept pattern formed by all the segments of songs in the dataset, and then construct FCP set and ICP set based on the frequency of concept patterns (refer to Sect. 3.3.3).

Algorithm 1: Infrequent Concept Pattern Filtering Process of a Segment

Input: $\mathcal{S}_{fcp}$: FCP set; $\mathcal{S}_{icp}$: ICP set; $\mathcal{C}$: Concept set of a segment;
$\mathcal{P}$: $P_c(c \in \mathcal{C})$ is the estimated probability of concept c in the segment
Output: $\mathcal{C}$; `// return the remaining concepts after filtering for the segment`

Form all the concept patterns $\mathcal{L}$ with $\mathcal{C}$;
$\mathcal{C}_{temp} = []$; `// define a empty set for concepts`
while $\mathcal{S}_{icp} \bigcap \mathcal{L} \neq \emptyset$ **do**
  for *each concept* $c \in \mathcal{S}_{icp} \bigcap \mathcal{L}$ **do**
    $\mathcal{C}_{temp} = \mathcal{C}_{temp} \bigcup c$;
  for *each concept* $c \in \mathcal{C}_{temp}$ **do**
    `/* count the number of times` c `in a ICP of the segment */`
    Get m_c: the number of concept patterns $l \in \mathcal{S}_{icp} \bigcap \mathcal{L}$ containing c ;
    `/* count the number of times` c `in a FCP of the segment */`
    Get n_c: the number of concept patterns $l \in \mathcal{S}_{fcp} \bigcap \mathcal{L}$ containing c ;
  `/* get the concepts which appear in the most number of ICPs */`
  Set $m = max(m_c, \forall c \in \mathcal{C}_{temp})$;
  $\mathcal{C}_{icp} = []$;
  for *each concept* $c \in \mathcal{C}_{temp}$ **do**
    if $m_c == m$ **then**
      $\mathcal{C}_{icp} = \mathcal{C}_{icp} \bigcup c$;
  `/* remove the concept which appears in the most number of ICPs */`
  if $|\mathcal{C}_{icp}| == 1$ **then**
    Remove $c \in \mathcal{C}_{icp}$ from $\mathcal{C}$;
  else
    `/* get the concepts which appear in the least number of FCPs */`
    Set $n = min(n_c, \forall c \in \mathcal{C}_{icp})$;
    $\mathcal{C}_{fcp} = []$;
    for *each concept* $c \in \mathcal{C}_{temp}$ **do**
      if $n_c == n$ **then**
        $\mathcal{C}_{fcp} = \mathcal{C}_{fcp} \bigcup c$;
    `/* remove the concept which appears in the least number of FCPs */`
    if $|\mathcal{C}_{fcp}| == 1$ **then**
      Remove $c \in \mathcal{C}_{fcp}$ from $\mathcal{C}$;
    else
      `/* if there are more than one concepts appearing in the most number of ICPs and the least number of FCPs, remove the ones with smallest probability */`
      Remove the concepts $c \in \mathcal{C}_{fcp}$ with the smallest $P_c(\forall c \in \mathcal{C}_{fcp})$ from $\mathcal{C}$;
  Re-form the concept patterns $\mathcal{L}$ with the remaining concepts $\mathcal{C}$;
  $\mathcal{C}_{temp} = []$;
Return $\mathcal{C}$;

- Step 3 - Noisy Concept Removal: For each segment of a song, detect the ICPs and remove suspicious concepts that cause such ICPs using Algorithm 1. Specifically, for the set of concepts in an ICP of a segment, we remove the one that appears in the most number of ICPs (line 9-15) or the least number of FCPs (line 17-23) in this segment. If two concepts appear in the same number of ICPs and FCPs (i.e., both concepts appear in the most number of ICPs and the least number of FCPs), the label with lower probability (P_{ij}) will be removed (line 25).

3.2.3 Location-aware Topic Model

In the real world, various songs could be suitable for a particular venue. A human possesses an amazing capability to judge *whether a song fits a venue* or *which song has higher suitability to a venue*. However, it is not easy to explicitly explain the reason in a straightforward way. Although people usually interpret music using various semantic concepts, explanation based on concepts or mixture of concepts could be inaccurate, less comprehensive and confusing in many cases. One approach is to describe and model a venue's characteristics via combining the musical concepts that are suitable for the venue. In other words, it maps the venue and music items into common musical concept space. The drawback of this method is lack of effective capability to model interactions between different concepts. Many music concepts are generally highly correlated and not independent of each other. In fact, they are intertwined together in a song to express certain semantics. For example, compiling the same song in different styles and using different instruments can create different atmosphere and give us different feelings. Music selection for a venue is highly related to the combinations and association of the multiple concepts. Motivated by these observation and discussion, we develop a novel topic model - Location-aware Topic Model (LTM) to facilitate a joint model-

ing of songs and venues under a latent topic space, in which the association and suitability between music and venues can be directly characterized and measured. In LTM, each latent topic is represented by a mixture of music concepts; in turn, songs and venues are the mixtures of topics. A topic of LTM can be treated as a particular interaction between music concepts. The topics and their associations (i.e., the representation of a venue) explain the underlying reasons why people prefer certain songs at a certain type of venue.

3.2.3.1 Model Description

Location-aware Topic Model (LTM) is a generative probabilistic model to characterize the associations between music contents and venue types. The associations are constructed via a set of latent semantic topics, which are discovered from a venue-labeled music corpus. The corpus consists of a set of songs labeled with one venue label or several venue labels, indicating that the song is suitable for these venues. The *common features* embedded in songs labeled with the same venue characterize the *music preference* of a venue. For the LTM, the music preference of a venue l is represented as a probabilistic distribution of latent topics, $\boldsymbol{\theta_l}$[4]. Meanwhile, each song s is also modeled as a probabilistic distribution of the same latent topics, $\boldsymbol{\theta_s}$, which captures the *latent semantics* expressed by the song. Each latent topic z is a probabilistic distribution of terms or music concepts , denoted as $\boldsymbol{\phi_z}$, which effectively captures rich interactions between different music concepts. LTM can be represented by the graphical model shown in Fig. 3.5. In the generation of a song s labeled with a venue l, for each word w_s of the song s, it could be generated based on the music preference of the venue $\boldsymbol{\theta_l}$, or generated according to this song's properties $\boldsymbol{\theta_s}$. As shown in the figure, LTM contains a switch mechanism which controls the generation of words based on the topic distribution of the venue

[4]Unless otherwise specified, notations in bold style denote matrices or vectors, and notations in normal style denote scalars

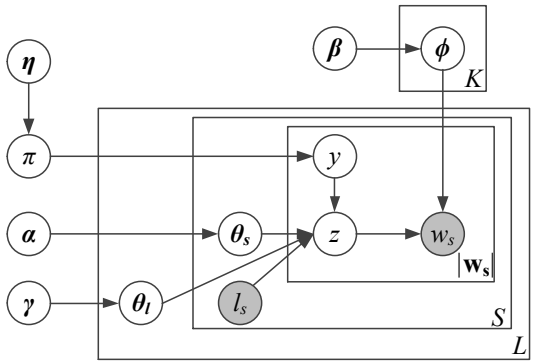


Figure 3.5: Plate notation of the Location-aware Topic Model.

l_s or the song s. In particular, an indicator variable $y \in \{0, 1\}$ from Bernoulli distribution parameterized by π associated with each word w_s. y acts as a switch: if $y = 0$, a topic z is drawn from $\boldsymbol{\theta_l}$ firstly, then word w_s is drawn from ϕ_z; otherwise, if $y = 1$, a topic z is drawn from $\boldsymbol{\theta_s}$ firstly, then word w_s is drawn from ϕ_z. Formally, the generative process of LTM is shown in Algorithm 2.

Algorithm 2: Generative Process of LTM

for *each topic* $z \in \{1, \cdots, K\}$ **do**
  Draw $\boldsymbol{\phi_z} \sim Dir(\cdot|\boldsymbol{\beta})$;
for *each song* $s \in \{1, \cdots, S\}$ **do**
  Draw $\boldsymbol{\theta_s} \sim Dir(\cdot|\boldsymbol{\alpha})$;
for *each venue* $l \in \{1, \cdots, L\}$ **do**
  Draw $\boldsymbol{\theta_l} \sim Dir(\cdot|\boldsymbol{\gamma})$;
for *each song* $s \in \{1, \cdots, S\}$ *labeled with a venue* $l_s \in \{1, \cdots, L\}^5$ **do**
  for *each word* $w_s \in \boldsymbol{w_s}$ *in the song* s **do**
    Draw $y \sim Bernoulli(\cdot|\pi)$;
    if $y == 0$ **then**
      Draw z from the topic distribution $\boldsymbol{\theta_{l_s}}$ of the venue l_s;
    if $y == 1$ **then**
      Draw z from the topic distribution $\boldsymbol{\theta_s}$ of the song s;
    Draw the word w_s from ϕ_z;

According to the generation process, the probability of a word w_s in a song s under venue type label l is:

$$\begin{aligned} P(w_s|s,l) &= \pi P(w_s|\boldsymbol{\theta_s}, \boldsymbol{\phi}, s) + (1-\pi)P(w_s|\boldsymbol{\theta_l}, \boldsymbol{\phi}, l) \\ &= \pi \sum_z P(w_s|z, \boldsymbol{\phi})P(z|\boldsymbol{\theta_s}, s) + (1-\pi)\sum_z P(w_s|z, \boldsymbol{\phi})P(z|\boldsymbol{\theta_l}, l) \end{aligned} \tag{3.2}$$

where $P(w_s|\boldsymbol{\theta_s}, \boldsymbol{\phi})$ is the probability that the word w_s in s is generated according to the song's music properties, $P(w_s|\boldsymbol{\theta_l}, \boldsymbol{\phi})$ is the probability that the word w_s in s is generated based on the venue's music preference. π is the Bernoulli parameter or *mixing weight* which controls the generation process. From the generation process, we can easily find that the topic distribution of a song is determined by the word (i.e., music concept) occurrences in this song. The generated latent topics are meant to capture the difference between songs. At the same time, the *word co-occurrence patterns* or *hidden associations between the words/concepts* embedded in the songs of a venue, are captured by the topic distribution of this venue. A venue's topic distribution can be regarded as the background distribution of the songs that are suitable for the venue and the topic distribution of each song is a variation of the venue's topic distribution. As different songs are suitable for different venues, the topics are also tailored for discriminating the characteristics of different venues.

The proposed LTM discovers (1) each venue's music distribution over latent topics $\boldsymbol{\theta_l}$, (2) each song's topic distribution $\boldsymbol{\theta_s}$, (3) topic distribution over music concepts $\boldsymbol{\phi}$, and (4) the mixing weight π. The generative model captures the associations between songs and venues via the generation of a venue-labeled music corpus. With the model hyperparameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}\}$, the generation probability of a corpus D with the observed and hidden variables:

$$P(D|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}) = \int \cdots \int \prod_{s=1}^{S} \prod_{i=1}^{|\boldsymbol{w_s}|} P(w_i|z, \boldsymbol{\phi}) P(\boldsymbol{\phi}|\boldsymbol{\beta}) P(z|\boldsymbol{\theta_s}, \boldsymbol{\theta_{l_s}}, y) \\ P(\boldsymbol{\theta_s}|\boldsymbol{\alpha}) P(\boldsymbol{\theta_{l_s}}|\boldsymbol{\gamma}) P(y|\pi) P(\pi|\boldsymbol{\eta}) d\boldsymbol{\theta_s} d\boldsymbol{\theta_{l_s}} d\boldsymbol{\phi} d\pi \tag{3.3}$$

3.2.3.2 Discussion

Here we discuss the differences of our proposed topic model with other topic models, including Labeled LDA [118], Author-Topic Model [144], and the location-aware topic model proposed in [157]. In Labeled LDA [118], the terms in a document are directly assigned to the labels of the documents, which indi-

cates that the latent topics of a document are limited to its labels. The Author-Topic model (ATM) [144] uses a topic-based representation to model both the contents of documents and the interests of authors. However, this model only focuses on the interests of authors while it cannot obtain the document-specific topic-mixture proportions. To use ATM in location-aware music recommendation, a location is treated as an "author", and all the songs labeled with the location are generated based on the topic distribution of the location. There are two limitations to use the method in location-aware music recommendation: (1) the model cannot capture the distinct characteristics of individual songs of a location, because these songs are all generated from the same topic distribution; and (2) for good performance, the ATM needs large numbers of "authors" to learn the latent topics. While in our context, a location refers to a type of venue, e.g., *library*. It is hard to collect enough data for thousands of venue types to learn such a model. The location-aware topic model in [157] was designed to explicitly model the relationship between locations and words. This model labels each word in a document with a location, but it cannot generate the topic distribution for a location. It is reasonable some textual keywords are related to a location, such as Personal Names ("Obama" is more likely related to US)or Regional Words ("CCTV" is more likely related to China)[6]. However, it is hard to relate a short segment of music (e.g., 1 second) to a certain place. Thus, with different design goals, the topic models discussed above are not suitable for location-aware music recommendation tasks. Different from these models, our proposed topic model can effectively discover the topic distributions of both songs and venues. Accordingly, the concepts relevant to venues and songs are mapped into the same latent space and can be directly matched in the space.

[6]Please refer to Table 2 in [157] for more examples

3.2.3.3 Model Inference

In the LTM model, the estimation of the generation probability of a corpus involves a set of parameters as shown in Eq. 3.3. Among them, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ are hyperparameters and pre-defined. The parameters to be estimated are (1) venue-topic distribution $\boldsymbol{\theta_l}$, (2) song-topic distribution $\boldsymbol{\theta_s}$, (3) topic-term distribution $\boldsymbol{\phi}$, and (4) Bernoulli distribution parameter π. Besides, in the generation process, we also need to assign the indicator vector $\boldsymbol{Y}$ and latent topic vector $\boldsymbol{Z}$ to the sequence of words $\boldsymbol{W}$ in the corpus. We apply collapsed Gibbs sampling to obtain samples of the hidden variable assignments and to estimate the unknown parameters $\{\theta_s, \theta_l, \phi, \pi\}$. In the collapsed Gibbs sampling, each latent variable is iteratively updated given the remaining variables. The parameters $\{\theta_s, \theta_l, \phi, \pi\}$ are estimated based on the results of a constructed Markov chain that converges to the posterior distribution on z. The Collapsed Gibbs Sampling process of LTM is described in Algorithm 3.

Algorithm 3: Collapsed Gibbs Sampling Process for LTM

Input: D: A venue-labeled music dataset;
K: number of topics;
Dirichlet priors: $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$;
Beta priors: $\boldsymbol{\eta}$

Output: Estimated parameters $\boldsymbol{\theta_s}$, $\boldsymbol{\theta_l}$, $\boldsymbol{\phi}$, π

Initialize $\boldsymbol{Z}$ and $\boldsymbol{Y}$ by assigning random values ;
Count N_l^k, N_s^k, and N_k^t based on initialized $\boldsymbol{Z}$;
Count N_{y_0} and N_{y_1} based on initialized $\boldsymbol{Y}$;
for *each Gibbs sampling iteration* **do**
    for *each song* $s = 1, \cdots, S$ **do**
        for *each word* $w_s = 1, \cdots, N_s$ **do**
            Sample $y_{w_s} \sim Bernoulli(\cdot|\pi)$ based on π's value computed by Eq. 3.7;
            if $y_{w_s} == 0$ **then**
                Draw z_{w_s} according to Eq. 3.8;
            if $y_{w_s} == 1$ **then**
                Draw z_{w_s} according to Eq. 3.9;
            Update N_l^k, N_s^k, and N_k^t based on $z_{w_s} = k$;
            Update N_{y_0} and N_{y_1} based y_{w_s} ;
Estimate model parameters $\boldsymbol{\theta_s}$, $\boldsymbol{\theta_l}$, $\boldsymbol{\phi}$, and π according to Eq. 3.6 and Eq. 3.7, respectively

Here we show how to joint sample $y_i \in \boldsymbol{Y}$ and $z_i \in \boldsymbol{Z}$ of a word $w_i \in \boldsymbol{W}$ conditioned on all other variables. y_i and z_i are needed to be sampled jointly, because y_i decides to whether sample z_i from $\boldsymbol{\theta_l}$ or from $\boldsymbol{\theta_s}$. Formally, we define that $\boldsymbol{W}$ is a sequence of words during the sampling process, $\boldsymbol{Z}$ and $\boldsymbol{Y}$ denote the set of topics z and indicators y to the word sequence, respectively. $\boldsymbol{W_{\neg i}}$ denotes $\boldsymbol{W}$ excluding the i-th word w_i. Similar notation is used for other variables. For $\boldsymbol{W} = \{w_i, \boldsymbol{W}_{\neg i}\}$, $\boldsymbol{Z} = \{z_i, \boldsymbol{Z}_{\neg i}\}$, and $\boldsymbol{Y} = \{y_i, \boldsymbol{Y}_{\neg i}\}$, the joint probability of sampling $z_i = k$ and $y_i = 0$ is:

$$\begin{aligned} &P(z_i = k, y_i = 0 | \boldsymbol{Z}_{\neg i}, \boldsymbol{Y}_{\neg i}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \\ &\qquad \propto (\eta_0 + N_{y_0, \neg i}) \cdot \frac{\gamma_k + N^k_{l, \neg i}}{\sum_{k=1}^{K}(\gamma_k + N^k_{l, \neg i})} \cdot \frac{\beta_t + N^t_{k, \neg i}}{\sum_{t=1}^{V}(\beta_t + N^t_{k, \neg i})} \end{aligned} \tag{3.4}$$

Similarly, the joint probability of sampling $z_i = k$ and $y_i = 1$ is:

$$\begin{aligned} &P(z_i = k, y_i = 1 | \boldsymbol{Z}_{\neg i}, \boldsymbol{Y}_{\neg i}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \\ &\qquad \propto (\eta_1 + N_{y_1, \neg i}) \cdot \frac{\alpha_k + N^k_{s, \neg i}}{\sum_{k=1}^{K}(\alpha_k + N^k_{s, \neg i})} \cdot \frac{\beta_t + N^t_{k, \neg i}}{\sum_{t=1}^{V}(\beta_t + N^t_{k, \neg i})} \end{aligned} \tag{3.5}$$

where N^k_l denotes the number of times observing topic k in venue l, N^k_s denotes the number of times observing topic k in song s, N^t_k denotes the number of times that term t observed with topic k. N_{y_0} and N_{y_1} denote the number of times that words are drawn from venues and songs, respectively. Based on the state of the Markov chain $\boldsymbol{Y}$ and $\boldsymbol{Z}$, we can estimate the parameters:

$$\theta_{s,k} = \frac{\alpha_k + N^k_s}{\sum_{k=1}^{K}(\alpha_k + N^k_s)} \qquad \theta_{l,k} = \frac{\gamma_k + N^k_l}{\sum_{k=1}^{K}(\gamma_k + N^k_l)} \tag{3.6}$$

$$\phi_{k,t} = \frac{\beta_t + N^t_k}{\sum_{t=1}^{V}(\beta_t + N^t_k)} \qquad \pi = \frac{\eta_1 + N_{y_1}}{\eta_1 + \eta_0 + N_{y_1} + N_{y_0}} \tag{3.7}$$

3.2.4 Discussion

In real application, the system should (1) be able to recommend new songs and (2) be updated/refined when more playlist information becomes available. In order to recommend new songs, the key problem is how to estimate the topic distributions of new songs. Since new songs do not have venue labels,

the topic distribution estimation in LTM is the same as the topic distribution estimation of new documents in LDA. Following the method described in [49] (see Section 7.1 in [49]), we first initialize the algorithm by randomly assigning topics to words (in the new songs) and then perform a number of loops through the Gibbs sampling updated locally for the words of new songs.

When more playlists become available, the online learning algorithms designed for LDA could be applied to our model, such as [52, 168]. As the collapsed Gibbs sampling is used in inference, we show how to extend the state of the Gibbs sampler to the new observations, i.e., $\{\boldsymbol{W}, \hat{\boldsymbol{W}}; \boldsymbol{Y}, \hat{\boldsymbol{Y}}; \boldsymbol{Z}, \hat{\boldsymbol{Z}}\}$. $\boldsymbol{W}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ denote the topic, indicator, and word sequences in original training corpus, respectively; and $\hat{\boldsymbol{W}}, \hat{\boldsymbol{Y}}$, and $\hat{\boldsymbol{Z}}$ denote the topic, indicator, and word sequences in the corpus of new playlists, respectively. We first initialize the algorithm by randomly assigning topics to words (of songs in the new playlists) and perform a number of loops through the Gibbs sampling updated locally for those words, Eq. 3.8 and Eq 3.9 become:

$$
\begin{aligned}
&P(z_i = k, y_i = 0 | \hat{\boldsymbol{Z}}_{\neg i}, \hat{\boldsymbol{Y}}_{\neg i}, \hat{\boldsymbol{W}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \\
&\propto (\eta_0 + N_{y_0,\neg i} + \hat{N}_{y_0,\neg i}) \cdot \frac{\gamma_k + N^k_{l,\neg i} + \hat{N}^k_{l,\neg i}}{\sum_{k=1}^{K}(\gamma_k + N^k_{l,\neg i} + \hat{N}^k_{l,\neg i})} \cdot \frac{\beta_t + N^t_{k,\neg i} + \hat{N}^t_{k,\neg i}}{\sum_{t=1}^{V}(\beta_t + N^t_{k,\neg i} + \hat{N}^t_{k,\neg i})}
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
&P(z_i = k, y_i = 1 | \hat{\boldsymbol{Z}}_{\neg i}, \hat{\boldsymbol{Y}}_{\neg i}, \hat{\boldsymbol{W}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \\
&\propto (\eta_1 + N_{y_1,\neg i} + \hat{N}_{y_1,\neg i}) \cdot \frac{\alpha_k + N^k_{s,\neg i} + \hat{N}^k_{s,\neg i}}{\sum_{k=1}^{K}(\alpha_k + N^k_{s,\neg i} + \hat{N}^k_{s,\neg i})} \cdot \frac{\beta_t + N^t_{k,\neg i} + \hat{N}^t_{k,\neg i}}{\sum_{t=1}^{V}(\beta_t + N^t_{k,\neg i} + \hat{N}^t_{k,\neg i})}
\end{aligned}
\tag{3.9}
$$

where $N_{y_0,\neg i}$, $N_{y_1,\neg i}$, $N^k_{s,\neg i}$, $N^k_{l,\neg i}$, and $N^t_{k,\neg i}$ are the previously obtained values on the original training data using Gibbs sampling(see Section 3.2.3.3); and $\hat{N}_{y_0,\neg i}$, $\hat{N}_{y_1,\neg i}$, $\hat{N}^k_{s,\neg i}$, $\hat{N}^k_{l,\neg i}$, and $\hat{N}^t_{k,\neg i}$ are the corresponding values observed in the new corpus. Then the parameters estimated by Eq. 3.6 and Eq. 3.7 are updated with the consideration of those new observations.

In fact, it would be better to train the model periodically. With the use of the system, a set of new songs and new playlists will be added into the corpus after a period. User's music preference on songs may evolve due to the evolution of music popularity trends. Re-training the model based on recent playlists could enable the system to track the recent music preference of users and always recommend songs that users prefer in recent time period. Besides, the model is trained in offline. With the recent techniques of training large topic models [167, 160], our model could be trained efficiently.

3.3 Experimental Setup

We conduct a series of experiments to study the performance of the VenueMusic System and try to address the following research questions:

RQ1 Is it better to use latent topics to capture the associations between songs and venues, comparing to the direct use of low-level audio features or semantic concepts?

RQ2 Is it better to represent songs as music concept sequences in the LTM than to represent songs as "bag-of-audio-words"?

RQ3 Does the use of Infrequent Concept Pattern Filtering (ICPF) process improve the final performance?

To answer these questions, we compare the performance of the Location-aware Topic Model (LTM) with four competitors: two content-based recommendation methods[7], a LTM based on "audio words" and a LTM based on the generated music concepts without the use of ICPF process on two datasets TC1.

[7]Collaborative-based filtering (CF) methods are not used in comparisons, because that CF is suitable for the cases with a large number of users (venues in our case), while there are only eight venues in our experiments.

3.3.1 Test Collection Construction

Test collection plays an important role in large scale performance evaluation and comparison. In this work, we carefully develop three test collections to facilitate empirical study.

3.3.1.1 Concept-Labeled Music Dataset

A dataset with songs labeled by music concepts is built for learning SVM classifiers to estimate the probabilities of music concepts in each music segment (described in Sect. 3.2.2.2). In experiments, three music concepts are used[8]: *genre*, *mood*, and *instrument*. The three types of concepts are selected because (1) they are important concepts usually used to describe music preferences according to studies from psychology and cognition [45, 120], and (2) they are the most commonly used music concepts to annotate songs by common users [76]. The *five* mood classes in MIREX mood classification task[9] are used in the *mood* dimension. Twelve genres are used in the *genre* dimension[10], and twelve instruments from four types of popular instruments [170] are used in the *instrument* dimension. The classes of each concept dimension are shown in Table 3.2. For each class, 100 songs were carefully selected. A 30 seconds audio stream for each selected song is downloaded from 7digital[11].

Details about the procedure of song selection for each concept dimension are described below.

- **Song Selection for *Mood*:** Songs for mood dimensions are collected from Allmusic[12], which is an expert-based music website. Allmusic pro-

[8] Although there are only three types of music concepts in our implementation, more concepts can be used. When more music concepts are used, our model is expected to model the song and venue more accurately.

[9] http://www.musicir.org/mirex/wiki/2009: Audio_Music_Mood_Classification

[10] According to the study of [120], fourteen general genres are well enough to represent user music preferences on the aspect of genre. The other two genres are *sound tracks* and *religious* besides the twelve genres used here.

[11] https://www.7digital.com/

[12] http://www.allmusic.com/

Table 3.2: Types of three music concepts used in experiments

Concept	Classes
Mood	aggressive, humorous, literate, passionate, rollicking
Genre	alternative, blues, classical, country, electronic, funk, hip-hop, jazz, metal, pop, reggae, rock
Instrument	trombone, trumpet, tuba, flute, clarinet, saxophone, piano, snare, drumkit, violin, cello, guitar

vides representative songs for various moods and genres.[13] There are 50 songs for each type. For the five classes of mood, each class represents a cluster of similar moods[14] (In Table 3.2, a mood represents a mood class.). The 50 songs provided by Allmusic for each mood in a mood class are collected first, and then 100 songs are randomly selected for the class.

- **Song Selection for *Genre*:** *Blues*, *classical*, *country*, *electronic*, *jazz*, and *reggae* are clearly listed in Allumusic, and provide 50 songs for each type. To obtain more songs of these genres and songs of other genres, we referred to DigitalDreamDoors[15], which provides more than 200 music & movie lists. These lists are created by a crowdsourcing method. The website allows people to review each list. Each list is revised regularly by the editor who creates the list based on users' comments. After collecting songs from corresponding genre lists in the website, three music hobbyists are asked to cross-check and select the songs for each genre. A song is selected for a certain genre when the three evaluators make an agreement. Through the above process, 100 songs are selected for each genre.

- **Song Selection for *Instrument*:** For each instrument, we search (1)

[13]http://www.allmusic.com/genres; http://www.allmusic.com/moods
[14]http://www.musicir.org/mirex/wiki/2009:Audio_Music_Mood_Classification
[15]http://www.digitaldreamdoor.com/pages/about_us_ddd.html. Access on 27 December 2013

albums and songs of famous soloists of the instrument, such as *Taylor Davis* for *violin*, *Alison Balsom* for *trumpet*, etc. and (2) search albums and songs using keywords like "*guitar solo*", "*guitar music*", "*guitar songs*" in 7digital. After collecting the candidate songs, the same assessment procedure as genre music selection is conducted to select 100 songs for each instrument. The selected songs of an instrument contain pure music, songs, solo and mixed with other instruments.

The selection procedure, which first selects songs from reliable resources and then manually checks the songs by human subjects, is helpful on guaranteeing the data quality and saving much time and labour. Notice that a relatively simple procedure is adopted to verify the *genre* and *instrument* of a candidate song. This is because: (1) in general, a song can be classified into a certain genre that majority will agree on, and (2) there is a definite answer to whether a song is played with a particular instrument or not. Because of the objective nature of the judgment on the genre and instrument of a song, it is easy for the subjects to achieve agreement on whether a song belongs to a genre or played with a particular instrument. Similar to the song selection procedure for each concept described above, candidate songs are first collected and then verified by human subjects.

3.3.1.2 Venue-Labeled Dataset (TC1)

In this dataset, each song is labeled with one or several venue types. The labels of a song indicate which venue types this song is suitable for. Eight representative types of venues in daily life are selected for the experiments. They include *library*, *gym*, *restaurant*, *bedroom*, *mall*, *office*, *bus/train*[16], and *bar*, where people often enjoy music. The song candidates for each venue were collected from the corresponding playlists in Grooveshark. Grooveshark contains a large amount of playlists created by users, titled with various contexts such as

[16]Bus and train are used to represent the *transportation*.

gym playlist, *bar music*, etc. These labeled playlists in GrooveShark have been successfully used for activity classification [159]. Venue-labeled playlists imply that users have special preferences on music contents in different venues, and also provide us a good source to collect data. In our implementation, for each venue, the playlists named by "*$venue$ songs*", "*$venue$ music*", and "*$venue$ playlist*" were retrieved in Grooveshark. Songs in the returned playlists were collected. Taken *bar* as an example, "*bar songs*", "*bar music*", and "*bar playlist*" were used to search related playlists. For each venue, we collected songs from at least 150 playlists. And more than 5000 individual songs were collected on average for each venue. Many songs appear in multiple playlists of a venue. For example, the song "*Nine Inch Nails - the hand that feeds*" appears in 48 playlists of *library*. As the playlists of a venue are created by different users, the appearance of a song in multiple playlists implies that people have similar preferences on music for a particular venue. Songs of a venue are sorted in descending order based on the number of playlists they appear in. The top 500 songs in the sorted list of each venue were selected.

The selected 500 candidate songs for each venue are then evaluated by human subjects. Nine subjects are volunteered for the evaluation. All the subjects are music hobbyists. They are five females and four males with different education backgrounds. Five of them are students, and the other four are working professionals. During the evaluation, they are required to listen to each song of a venue and then rate the song. The guidelines of rating are shown in Table 3.3. The subjects need to listen to a song for at least 60 seconds before making the final decision.

We studied the inter-subject agreement by calculating Fleiss's Kappa [78] among the 9 subjects for every venue. All Kappa values are significantly higher than 0 (p-value < 0.001) with the lowest value for *restaurant* (0.076) and *mall* (0.09) and especially high for *bedroom* (0.337), *gym* (0.314) and *library* (0.287). The average Kappa value over eight venues is 0.202 ($\pm$0.100). The results

Table 3.3: Guidelines of rating a song for a type of venue

Score	Description
1 point	I absolutely will not listen to it in this type of venue
2 point	I can stand it in this type of venue
3 point	I do not mind to listen to it in this type of venue
4 point	I like it in this type of venue
5 point	I like it very much in this type of venue

Table 3.4: Number of relevant songs for each venue in TC1

Bar	Gym	Library	Office	Restaurant	Mall	Bus/Train	Bedroom
266	233	154	176	121	135	221	189

indicate that subjects have statistically significant agreement on music for venues. To evaluate the precision and ranking performance of the methods, the ratings of a song for a venue are converted into three relevance levels. Specifically, if the majority of the subjects (namely, 5 or more subjects) give a rating greater than 3 point to a song for a particular venue, then the song is regarded as *relevant* for the venue; if the majority of the subjects give a rating less than 3 point to a song for a particular venue, then the song is regarded as *irrelevant* for the venue; if a song does not belong to either relevant or irrelevant, it is regarded as *neutral*. The number of relevant songs for each venue is shown in Table 3.4.

3.3.1.3 Large Music Dataset (TC2)

Since TC1 is relatively small, another test collection (TC2) was developed for large scale evaluation. TC2 contains 10,000 popular music selected from Last.fm[17]. This collection is constructed as follows. Artists from the top 150 artists in each week (namely, the most popular 150 artists in each week) from 20 February 2005 to 24 November 2013 in the category of *all places*[18] were collected in Last.fm. As the data in Last.fm is known to contain misspellings

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

[18]http://www.last.fm/charts/artists/top/place/all?limit=150.

and mistakes, the collected artist list was checked by matching each artist name in AllMusic. After filtering, the list contains 531 artists. The songs of each artist were collected from the MusicBrainz database[19]. For the songs in Last.fm, we collected the number of its listeners till 26 November, 2013, when accessing the data. Finally, the top 10,000 songs with more listeners were obtained and their audio tracks are downloaded from 7digital.

3.3.2 Competitors and Evaluation Metrics

In the following presentation, we use **CLTM_F** to represent our proposed method, which uses LTM based on the extracted music concept sequence with the infrequent concept pattern filtering. We present the results of the following four competitors with CLTM_F to study the four research questions mentioned above.

- **Audio-Based Filtering (ABF)** Each venue is represented by several representative audio feature vectors. Specifically, by representing the songs of a venue using the audio features described in Sect. 3.2.2.1, K-means method is applied to generate k clusters. The feature vectors of the cluster centers are then used to represent the venue. The similarity between a representative vector of a venue and the feature vector of a new song is calculated by Euclidean distance. The best performance over these representative vectors of a venue is used to compare with other methods.

- **Concept-Based Filtering (CBF)** In this method, the histogram of music concepts is used to represent songs and venues. Specifically, based on the generated music concept sequence of a song (described in Sect. 3.2.2), the occurrence times of music concepts in the signature are counted and normalized to generate a histogram vector, which is used to represent the

[19]http://musicbrainz.org/. Access on 24 November, 2013

song. By aggregating all the music concepts of all songs of a venue, the concept histogram of the venue can be obtained. Then the KL distance is used to compute the similarity between songs and the venue.

- **Audio Word based LTM (ALTM)** This method uses "*bag-of-audio-words*" as input in the LTM. Specifically, each song in a corpus is segmented into small frames, and audio features are extracted from each frame. K-means method is used to group the frames into clusters based on their audio features. The cluster centers are used as "audio words". Indexing each frame of a song with the closest "audio words", the song is represented as a sequence of audio words. In our implementation, an audio word is a 0.5s music frame.

- **CLTM** Comparing with CLTM_F, this method doesn't have the module to support the infrequent concept pattern filtering process.

Besides the competitors above, we also compared LTM with other methods, such as Jaccard Similarity in [19][20], Autotagger[21] and two LDA variants (i.e., Author-Topic Model [144] and the location-aware topic model in [157]). Because these methods are not designed for current tasks - recommend songs to venue types[22], their performances are very limited[23]. Thus, we only present the results of four competitors listed above.

Precision at k (Precision@k), Average Precision at k (AP@k) and Normalized Discounted Cumulative Gain at k (NDCG@k) [57] are used as evaluation metrics. Please refer to Appendix A for the descriptions about these metrics.

[20]In [19], a song and a Point of Interest are matched based on the similarity between manually labeled concepts. In our implementation, as no manual labels is available, we use the generated concept vectors of songs and venues (concept generation of venues is described in Sect. 3.3.2) for computing Jaccard similarity.

[21]Autotagger is used to classify each song into different venues.

[22]The reasons of the two topic models are described in Sect. 3.2.3.2; Jaccard Similarity in [19] relies on manually labeled tags; and Autotagger is a classification methods.

[23]For all the four methods, their average accuracies (precision@20) in TC2 are lower than 20%. The highest precision for the four methods are obtained by Jaccard similarity: 0.1875

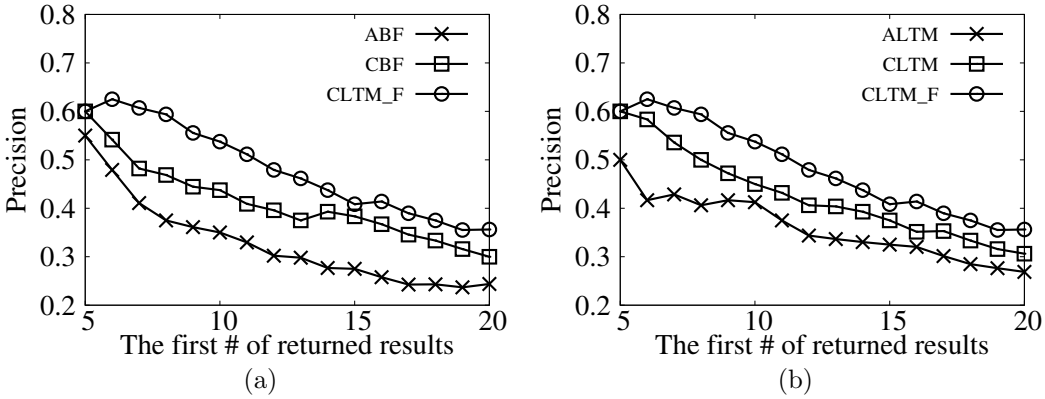


Figure 3.6: Average precision@$\{5-20\}$ comparison of different methods on Test Collection 1 (TC1).

3.3.3 Experimental Configurations

In our experiments, TC1 is split into training set and test set. Specifically, for each venue, 70% relevant songs were randomly selected to construct the training set. The test data contains 1000 songs, comprised by the rest 30% relevant songs of each venue and randomly selected 552 songs from the rest songs of TC1 (excluding the relevant songs of venues). The representations of venues (for ABF and CBF methods) are obtained based on the songs in the train set. ALTM, CLTM and CLTM_F are also trained based on the train set of TC1. The learned models based on the train set of TC1 are directly used in TC2. We focus on the performance improvement achieved by CLTM_F over other methods. For details about the parameter setting in our experiments, please refer to [33].

3.4 Experimental Results

3.4.1 Performance Evaluation on TC1

In this section, we compare and analyze the performance of five methods on TC1. Fig. 3.6a shows the average precision@ $\{5-20\}$ of recommendations using acoustic features (ABF), concept histogram (CBF), and our method

(CLTM_F). Comparisons between them are to verify the advantages of using the LTM generated topic distributions to represent songs and venues (**RQ1**). From the figure, we can observe that CBF is consistently better than ABF, and CLTM_F clearly outperforms ABF and CBF with statistically significant improvement. The results demonstrate that low-level acoustic features alone cannot well represent the associations between music contents and venues. The generated topics by the LTM associating the music concepts in high-level semantic space can better capture the connections between music contents and venues. The comparisons between ALTM, CLTM and CLTM_F shown in Fig. 3.6b demonstrate the advantages of learning topics using music concepts over audio words (**RQ2**) and the usefulness of ICPF (**RQ3**). The ICPF process indeed improves the final performance, as CLTM_F outperforms CLTM. It implies the process can obtain more suitable music concept sequence for a song. Furthermore, by comparing the results in Fig. 3.6a and Fig. 3.6b crossly, we observe that the performance of ALTM is only comparable to that of ABF, and CLTM is slightly better than CBF. The results indicate that the quality of music representation is crucial to the success of our LTM on venue-aware music recommendation.

Table 3.5: Precision and Average Precision comparison across different venues on Test Collection 1 (TC1)

Venue	Precision@20					AP@20				
	ABF	ALTM	CBF	CLTM	CLTM_F	ABF	ALTM	CBF	CLTM	CLTM_F
Bar	.300	.300	.350	.300	**.400**	.118	.226	.173	.197	**.317**
Bedroom	.250	.350	.350	.400	**.450**	.189	.151	.297	**.307**	.301
Gym	.250	.350	.350	.400	**.450**	.104	.216	.325	.294	**.326**
Library	.200	.200	.250	.250	**.300**	.117	.115	.151	.144	**.176**
Office	.150	.200	.300	.250	**.450**	.057	.085	.141	.169	**.259**
Restaurant	**.300**	.150	**.300**	.250	.250	.151	.053	**.220**	.089	.105
Mall	.200	.200	.200	.200	.200	.077	.073	.069	.118	**.120**
Bus/Train	.300	**.400**	.300	**.400**	.350	.169	.273	.116	**.288**	.272
Mean	.244	.269	.300	.306	**.356**	.123	.149	.187	.201	**.235**

Table 3.5 shows the *Precision@20* and *AP@20* of our methods and other

competitors in each venue. CBF achieves better results than ABF method in *bar*, *bedroom*, *gym*, *library*, and *office*, while does not show any improvement in *restaurant*, *mall*, and *bus/train*. This is an interesting observation, which is in accord with the inter-person annotation agreement analysis (Sect. 3.3.1.2). A possible explanation is that people may have more consistent preferences on the types of music they like in the former five venues than the later three venues. *Bar* and *gym* have their special atmosphere where people tend to enjoy certain types of music. For these venues, the performance can be further improved by CLTM_F. It implies that for the venues where music concepts can directly describe on some extent, the CLTM_F topics can better capture the semantics of the venues. Similar to CBF, CLTM_F does not show any improvement in *mall*, and even performs worse in *restaurant*. This is partially because there are different kinds of *mall* and *restaurant*, subjects annotated songs based on the types of mall and restaurant they frequently visit in daily life. Accordingly, the obtained relevant songs for the two venues are relatively diverse (low Kappa values). Consequently, it is harder for the model to capture the associations between music and the two venues, resulting in poor performance. Particularly, CLTM and CLTM_F achieve better results in *bus/train*, where the CBF does not show any advantages over ABF. This suggests that music concept based LTM methods have the potential to capture the underlying reasons for music preference in the venue where the music concepts cannot well explain (**RQ1**). Comparing with CLTM, CLTM_F demonstrates much more consistent performance, which clearly shows the effectiveness of infrequent concept pattern filtering (**RQ3**).

To evaluate and compare the ranking performance of the methods, NDCG@20 are calculated for all methods and presented in Table 3.6. CLTM_F achieve the best result over the other methods at venues *bar*, *bedroom*, *gym*, *library*, and *office*. CLTM performs better than the other three methods on all venues except *restaurant* and *mall*. The results show that the superiority of music

concepts based LTM methods on finding the most suitable songs for venues (**RQ1** and **RQ2**).

Table 3.6: NDCG@20 comparison of different methods across different venues on Test Collection 1 (TC1)

Venue	ABF	ALTM	CBF	CLTM	CLTM_F
Bar	.291	.428	.349	.400	**.524**
Bedroom	.381	.330	.492	.515	**.527**
Gym	.272	.432	.506	.508	**.534**
Library	.295	.293	.345	.306	**.385**
Office	.173	.222	.308	.358	**.456**
Restaurant	**.423**	.169	.361	.270	.331
Mall	.295	.221	.206	.225	**.296**
Bus/Train	.380	**.507**	.299	.496	.477
Mean	.314	.325	.358	.385	**.441**

3.4.2 Performance Evaluation on TC2

We observed the achieved improvements of CLTM_F on music recommendation for specific venues in TC1, while TC1 is a weakly-labeled dataset[24]. To validate the real performance of the method on a large dataset, we evaluate its performance and compare it with other competitors on TC2. The results returned by each method are carefully evaluated by human subjects. Specifically, the five methods were used to recommend songs from TC2 for the eight venues. The top 20 recommended songs are collected and mingled together to form a single playlist for a venue. The optimal models of ALTM, CLTM and CLTM_F obtained on TC1 were used. To fairly evaluate whether the songs in a playlist are suitable for the corresponding venue, 7 human subjects were recruited. They are 4 females and 3 males with different education background from Singapore and China (They are a different set of subjects from the subjects for TC1 annotation). The subjects are required to listen to the

[24]It is possible that a song is suitable for a venue, while has not been labeled for the venue.

recommended songs in the corresponding venues[25] and rate them according to the rule described in Sect. 3.3.1.2. Each subject is required to assess each song in all playlists. With the collected ratings, each song in the results of a venue is judged as relevant, neutral and irrelevant using the same method described in Sect. 3.3.1.2. Based on the relevance judgment of each song in the playlists for venues, *Precison@20*, *AP@20* and *NDCG@20* are computed for each method in each venue. The results are shown in Table 3.7 and Table 3.8.

Table 3.7: Precision and Average Precision comparison across different venues on Test Collection 2 (TC2)

Venue	Precision@20					AP@20				
	ABF	ALTM	CBF	CLTM	CLTM_F	ABF	ALTM	CBF	CLTM	CLTM_F
Bar	**.950**	.800	.850	**.950**	**.950**	.845	.746	.723	.906	**.950**
Bedroom	.250	.450	.450	.550	**.650**	.169	.245	.339	.347	**.491**
Gym	.400	.350	.450	.550	**.650**	.189	.179	.248	.428	**.515**
Library	.350	.300	.600	.600	**.650**	.257	.154	.344	.437	**.452**
Office	.400	.350	.450	.450	**.500**	.175	.162	.213	.235	**.269**
Restaurant	**0.30**	.150	.200	**.300**	**.300**	.083	.097	.128	.097	**.156**
Mall	.150	.250	.350	.300	**.450**	.024	.072	.158	.102	**.200**
Bus/Train	.200	.200	.500	.450	**.550**	.067	.047	.217	.359	**.367**
Mean	.375	.356	.481	.519	**.588**	.226	.213	.296	.364	**.425**

Table 3.8: NDCG@20 comparison of different methods across different venues on Test Collection 2 (TC2)

Venue	ABF	ALTM	CBF	CLTM	CLTM_F
Bar	.910	.853	.859	.955	**.968**
Bedroom	.365	.451	.555	.541	**.700**
Gym	.393	.402	.485	.619	**.706**
Library	.456	.335	.533	.538	**.625**
Office	.383	.389	.456	.438	**.474**
Restaurant	.251	.253	.308	.272	**.367**
Mall	.117	.227	.382	.275	**.402**
Bus/Train	.214	.172	.425	.568	**.586**
Mean	.386	.385	.500	.526	**.604**

[25]We did not specify the exact location for each venue. They can go to the venues they usually go to.

From the results of *Precision@20* and *AP@20*, we can see that CLTM methods (CLTM and CLTM_F) achieve more than 50% recommendation accuracy for *bar*, *bedroom*, *gym*, *library* and *bus/train*, and significant improvement over other methods in these venues expect *bar*, where all methods can achieve high recommendation accuracy. Comparing to CLTM, CLTM_F presents more consistent performance and outperforms other methods across all venues, which implies the necessity of removing noisy concepts. As shown in the Table 3.8, CLTM_F outperforms other methods in all venues on ranking performance, and achieves significant improvement in *bedroom*, *gym*, *library*, *office* and *bus/train*. The overall performance of CLTM is better than the other three methods, while its performance is not stable as CLTM_F across different venues. The performances in *restaurant* and *mall* are still unsatisfactory. As subjects judged the results based on the venues they went to, it is possible that the recommended songs are suitable for other types of malls or restaurants. For further study of the problem, it is necessary to classify the venues into finer granularity, such as specify the types of mall and restaurants.

3.5 Summary

In this chapter, we present a location-aware music recommender system called VenueMusic. This system can effectively recommend suitable songs for common venues in daily life. We have detailed a Location-aware Topic Model, which represents the music profiles of venues in a latent semantic space. A process of generating high quality music concept sequence for songs was described. The generated music concept sequences can effectively learn LTM for recommending songs for various types of venues. Two large datasets were constructed to evaluate the performance of our system. Experimental results demonstrate the effectiveness of our system.

Chapter 4

User Information Aware Text-Based Music Retrieval

In this chapter, we present a user information aware text-based music retrieval system. The goal of the system is to leverage the easily obtained user information (i.e., age and gender) to improve the search accuracy. Thus, the system can be used to deal with the cold-start problem of new users in personalized music retrieval systems. In fact, user-specific information, such as age and gender, has great influence on personal music preferences and interests. However, the existing research pays few attentions on designing advanced schemes for modeling and integration of user specific information to facilitate text-based music retrieval. By analyzing large-scale users' music profiles in Last.fm, we observe the influence of age and gender on music preference. Based on the observations, a novel topic model based scheme called User-Information-Aware Music Interest Topic (UIA-MIT) model is proposed to capture the influence of user's age and gender on user's music preferences. Further, by capturing the correlation of user's music preference, song, and semantic tags in a latent music interest space, we develop a user information aware retrieval framework, which can search and re-rank the search results based on age- and/or gender-specific music preference. An extensive experimental study demonstrates the

superiority of our method over the state-of-the-art text-based music retrieval methods from various perspectives.

4.1 Introduction

With the fast development of mobile computing technology and cloud-based streaming music service, personal handheld devices have been becoming the most popular platform to consume music daily. Based on Nelsen's Music 360 2015 report, 44% of US music listeners use smartphones to listen to music in a typical week. Typically, smartphones are for personal use. Thus, it is easy to obtain personal information via smartphones, which can be used in personalized applications to achieve better user experiences. With the fast growing trend in music consumption with smartphones, there has been an increasing interest in the multimedia database and information system communities to study the technology for supporting user-centered music information retrieval. Techniques for effective user-centered music search are gaining in its importance due to a wide range of potential applications. Based on this technology, personalized music search or recommendation systems can be developed to automatically cater for users' music needs.

Generally, user's music selection is greatly influenced by long-term music interests, which is dependent on user-specific background, such as age, gender, social status, grow-up environment, culture background, etc. Fig. 4.1 illustrates the influences of age and gender on user's music preferences. Based on a large number of real users' profiles from Last.fm[1], users are classified into different 14 age and gender groups. For example, "16-20_male" refers to the group of male users in ages of 16 to 20 years old. Similar definitions can be generalized to other groups. This figure presents the percentage of different artists in top 20, 50 and 100 favorite artists between 16-20_male users with

[1]Please refer to Section 4.3.1 for details about the users' profile.

other 13 group users [2]. From the figure, we can clearly observe that (1) users with larger age difference have more different favorite artists, and (2) the tastes of users in the same gender are more similar than users in different genders. For example, users in 16-20_male group and 21-25_male group have more common favorite artists than users in 16-20_male group and 21-25_female group. The observations demonstrate the importance of age and gender on listeners' music preferences.

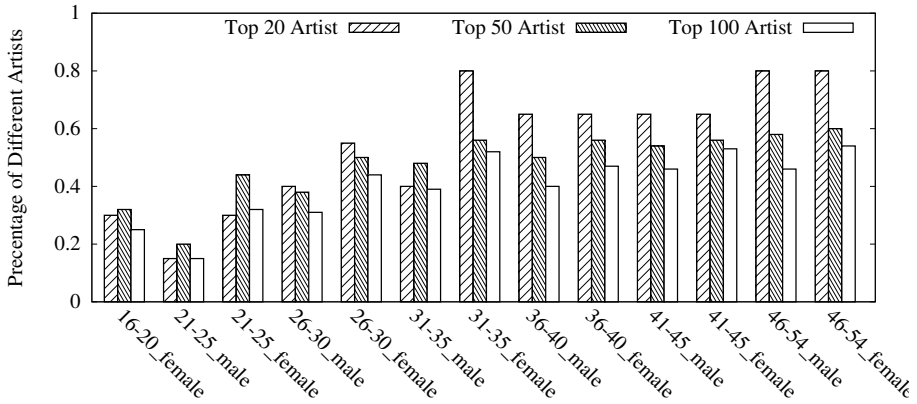


Figure 4.1: Percentage of different artists in top 20, 50, and 100 favorite artists between the 16-20_male group and other groups. The percentage of different artists in top K is the number of different artists between two groups in the top K artists divided by K.

Semantic-based music retrieval, as one of the most popular music search paradigms, typically requires users to provide a few text keywords as query to describe their music information needs. Because of the great impact of user-specific information on music preferences, given the same query, different users would expect different search results. For example, given a query "pop, sad", the retrieved songs expected by 40 years old male could be very different from the ones favored by 20 years old female. Here, *user-specific information* refers to the user related information which could influence user's long-term music interests. Despite the importance, user-specific information have not been well

[2]The most favorite artist of a user group is the one whose songs are loved by the most number of users in this group. In this figure, the 16-20_male group is used as the comparative base to compare with other groups. The same conclusions can be observed using other groups as comparative basis.

explored in existing music retrieval systems. Few studies focus on designing advanced schemes to exploit user-specific information in music retrieval. In this study, we develop a text-based music retrieval system, which can leverage user-specific information to improve the search performance. Because user-specific information is relatively easy to be obtained, especially on the mobile platform, the system has a wide range of real applications. Besides, it can be applied to deal with the cold-start problem of new users in personalized music retrieval system.

Two main challenges of integrating the user-specific information in music retrieval are (1) how to model the influence of user-specific information on music preferences and (2) how to associate the influence with query and songs. To tackle the challenges, a novel topic model called User Information Aware Music Interest Topic (UIA-MIT) model is proposed. UIA-MIT can explicitly model the music preferences of different types of user-specific information. We focus on age and gender information in this study. In this model, the music preferences depending on different factors (i.e., age and gender) are represented by the probabilistic distributions of a set of latent topics. The latent topics are represented by the probabilistic distributions of songs and terms (song's annotations or tags). Therefore, song, term, and the music preferences (influenced by age and gender) are associated via the latent topics. Based on the UIA-MIT model, we develop a probabilistic semantic-based music retrieval method, which aims at effectively satisfying user's personal music information needs by taking user-specific information into account.

In order to evaluate the method's performance and demonstrate advantages of our proposed method, we have conducted a series of experimental studies and comprehensive comparisons over different methods on two different search related tasks: ad-hoc retrieval and re-ranking. Our empirical results show age or/and gender information play important roles in search performance improvement. It demonstrates the importance and practical meaning of utilizing

user-specific information in real music retrieval system development. To the best of our knowledge, our work is the first attempt on designing advanced music retrieval methods to leverage user-specific information in retrieval.

The remainder of this chapter is organized as follows: In Section 4.2, we describe the UIA-MIT model and introduce the related retrieval methods. Section 4.3 presents experimental configuration and Section 4.4 gives a detail descriptions on experimental results. Finally, Section 4.5 concludes this chapter with a detailed summary.

4.2 User Information Aware Music Retrieval System

In this section, we describe the user information aware music retrieval system, which consists of two components:(1) a User Information Aware Music Interest Topic (UIA-MIT) Model, which captures the influence of user information (age and gender in this study) on user's music preference; and (2) a user information aware music retrieval method, which could leverage user's age and gender information in music retrieval based on the captured influence of age and gender on music preference.

4.2.1 Music Interest Discovery Topic Models

In this section, we present two topic models - Music Interest Topic (MIT) model and User-Information-Aware Music Interest Topic (UIA-MIT) model, which aim at capturing the *latent music interests* of users underlying the observations of (user, song) records. In both models, a set of latent topics (i.e., K topics) is discovered based on the records of users' loved tracks. Each latent topic represents one type/style of music or a *music interest dimension*. Notice that the music interest of a user is influenced by many factors, such as personality,

age, gender, country, etc. The MIT model has not explicitly modeled the influence of those factors and represents the latent music interest of a user as a probabilistic distribution of the latent topics. On the other hand, the UIA-MIT model is to capture the influence of different factors (e.g., age and gender) on music interests. For example, what is the *general music interests* of users *in a certain age range or gender*; or in other words, the *likelihood* of each type of songs loved by the users with regard to their ages and genders. This kind of knowledge can help us refine the search results in music retrieval. In the UIA-MIT model, a user's latent music interest is expressed as a mixture of multiple latent topic distributions. Each latent topic distribution represents the music interests depending on a factor (e.g., age). Therefore, the mixture of multiple latent topic distributions in this model represents a user's latent music interests, as the result of collective effects of different factors. In the following, we first introduce the key concepts (Sect. 4.2.1.1), and then describe the MIT (Sect. 4.2.1.2) and UIA-MIT model (Sect. 4.2.1.3), as well as the algorithm to inference the model (Sect. 4.2.1.4).

4.2.1.1 Preliminary

For ease of understanding and presentation, we first introduce some key concepts and notations.

Dataset D. The dataset D in our model learning consists of user, user information (i.e., age and gender), song, and song's content (i.e., tags and audio words), that is, $(u, a, g, s, \boldsymbol{s_w}, \boldsymbol{s_v}) \in D$, where $u \in \mathcal{U}$, $s \in \mathcal{S}$, $a \in \mathcal{A}$, $g \in \mathcal{G}$, $s_w \in \mathcal{W}$, and $s_v \in \mathcal{V}$. One piece of record in the dataset is that a user u of age a and gender g loves a song s with tags $\boldsymbol{s_w}$ and audio content $\boldsymbol{s_v}$: $D_{usc} = \{(u, a, g, s, w, v) : w \in \boldsymbol{s_w}, v \in \boldsymbol{s_v}\}$. The corpus is formulated into three types of documents: (1) user-song-document: it is the user's music profile, which is the sequence of loved songs of the users. Each song represents a "word" in this type of document; (2) song-word-document: this type of document represents

the semantic content of a song. For each song, its text word document is formed by firstly parsing its social tags (in Last.fm) into individual words and then concatenating those words; (3) song-audio-document: this type of document represents the audio content of a song. For each song, it is comprised by the audio words of the song. The implementation details of the three types of documents in our study are described in Section 4.3.

Audio Word. An audio word is a representative short frame (e.g., 23ms in our implementation) of the audio stream in a music corpus [127, 147]. Audio words are used to represented the audio content of a song into a "bag-of-audio-words" document. The general procedures to generate the audio words are: (1) segment the audio stream of each song in a corpus into short frames; (2) extract acoustic features from each short frame; (3) use a clustering algorithm (e.g.,k-means) to cluster the short frames into n clusters based on their acoustic features. The cluster centers are the generated audio words for the corpus. By encoding each short frame of a song by the nearest cluster center (or audio word), then the song is indexed as a sequence of audio words, namely, a "bag-of-audio-words" document.

User Profile. For each user u in the dataset D, we create a user profile D_u, which including user's age and gender (i.e., age and gender) as well as loved tracks (i.e., $\{u, a, g, s\} \in D_u$).

Latent Topic. A latent topic z, or topic for short, in a song collection $\mathcal{S}$ is presented by a topic model $\boldsymbol{\phi_s}$, which is a probabilistic distribution over songs, that is, $\{P(s|\boldsymbol{\phi_s}) : s \in \mathcal{S}\}$ or $\{\phi_{k,s} : z = k, s \in \mathcal{S}\}$. Similarly, a topic in a text word corpus $\mathcal{W}$ is represented by a topic model $\boldsymbol{\phi_w}$, which is a probabilistic distribution over text words, that is $\{P(w|\boldsymbol{\phi_w}) : w \in \mathcal{W}\}$ or $\{\phi_{k,w} : z = k, w \in \mathcal{W}\}$. A topic in a audio word corpus $\mathcal{V}$ is represented by a topic model $\boldsymbol{\phi_v}$, which is a probabilistic distribution over audio words, that is $\{P(v|\boldsymbol{\phi_v}) : v \in \mathcal{V}\}$ or $\{\phi_{k,v} : z = k, v \in \mathcal{V}\}$.

Age and Gender Music Preferences. In the UIA-MIT model, we

categorize users into groups based on their age and gender information. As users in similar ages tending to have similar music interests, we categorize users in similar ages into an age group (presented in Sect. 4.3.1). The age music preference denotes *the music preferences of a certain age range a* or *the general music preferences of users in the age range a*, represented by $\boldsymbol{\theta_a}$, a probabilistic distribution over topics. Similarly, the gender music preference denotes *the music preferences of gender*, denoted by $\boldsymbol{\theta_g}$, a probabilistic distribution over topics.

User's Music Interest. In both MIT and UIA-MIT models, a user's music interest, denoted as $\boldsymbol{\theta_u}$, is a probabilistic distribution over topics. The user's music interest can be influenced by many factors (i.e., personality, age, gender, country, etc.). MIT models a user's music interest without explicitly modeling any individual factors, thus the user music interest $\boldsymbol{\theta_u}$ in MIT denotes user's personal music interest as the results influenced by all factors. UIA-MIT explicitly models the music preferences of ages and genders. In UIA-MIT, a user's music interest is comprised by the mixture of three topic distribution (see Eq. 4.1): (1) $\boldsymbol{\theta_u}$: the music preferences as a collective results based on the influences of on all other factors (e.g., personality) besides age and gender , (2) $\boldsymbol{\theta_a}$: age music preference or the music preference with regard to user's age , and (3) $\boldsymbol{\theta_g}$: gender music preference or the music preference with regard to gender. For the simplicity of presentation, in UIA-MIT, we also call $\boldsymbol{\theta_u}$ as user's music interest.

4.2.1.2 Music Interest Topic Model

Given a corpus of a large number of users and their loved tracks, user's latent music interest can be discovered using latent factor models, such as matrix factorization [73] and topic models. The proposed *M*usic *I*nterest *T*opic (MIT) model is an extension of Latent Dirichlet Allocation [15]. LDA is used to discover the latent topics or themes of a text document corpus. Similarly,

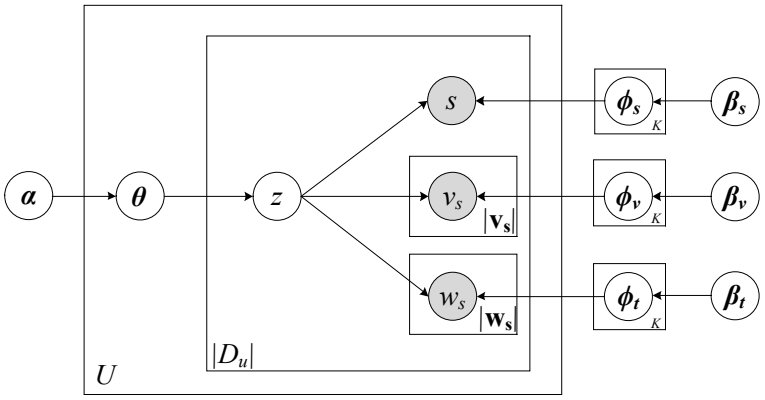


Figure 4.2: The graphical representation of the MIT model. Following the standard graphical model formalism, nodes represent random variables and edges indicate possible dependence. Shaded nodes are observed random variables.

MIT discovers the latent music interest dimensions underlying a user-song corpus, which contains users' loved songs and the contents of songs. In LDA, the latent topics are discovered based on the co-occurrence patterns of words in the documents. In MIT, the music interest dimensions are discovered by mining the music content co-occurrence patterns in users' loved songs.

The graphic representation of this model is shown in Fig. 4.2. Given a loved song s of a user u, it is assumed to be generated by first choosing a topic $z \in \mathcal{Z}$ from the user's music interests $\boldsymbol{\theta_u}$; then the song is sampled according to the song distribution $\phi_{k,s}$ of the chosen $z = k$. At the same time, its contents (i.e., text words $\boldsymbol{s_w}$ and audio words $\boldsymbol{s_v}$) are generated according to $\phi_{k,w}$ and $\phi_{k,v}$, respectively. The model captures the songs' associations based on their co-occurrences in the same user's profile and the word (both textual and audio words) associations based on their co-occurrences in the same song. Therefore, the model discovers a latent music space based on the co-occurrences of songs and their contents. A topic z is represented as a distribution of songs, a distribution of text words and a distribution of audio words, respectively. Because the social tags of songs could be noisy (e.g., unrelated and incomplete), the consideration of audio contents is helpful on generating meaningful latent topics (see Eq. 4.3, Eq. 4.5 and Eq. 4.6).

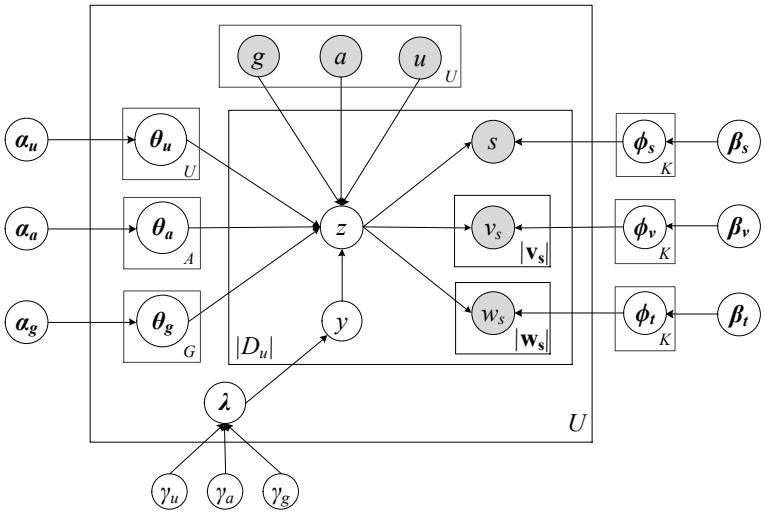


Figure 4.3: The graphical representation of the UIA-MIT model.

4.2.1.3 User Information Aware Music Interest Topic Model

In general, users with similar demographics have more similar music interests than users with different demographics. For example, users in the same age or gender have more similar music interests [12, 81, 55, 80]. To model the influence of user-specific factors, we propose a User Information Aware Music Interest Topic (UIA-MIT) model, which extends MIT to model user's music interest as a mixture of the music preferences (represented as the probabilistic distributions) depending on different factors. The graphic representation of the model is shown in Figure 4.3. As shown in the figure, UIA-MIT explicitly models the music preferences of ages ($\boldsymbol{\theta_a}$) and genders ($\boldsymbol{\theta_g}$). The music preference, as a result of all other factors (such as user's personality and country), is modeled as a single probabilistic distribution of latent topics, denoted as user's personal music interest ($\boldsymbol{\theta_u}$). The user's music interest ($\boldsymbol{\theta_u}$) in MIT can be regarded as the topic distribution as the collective result of all the influence factors, including age, gender, and other factors. Notice that the UIA-MIT model can be extended to model the music preference of other individual factors.

From the generation perspective, the model mimics the music selection process by considering the user's music interest, age music preference, and

gender music preference in a unified manner. Given a user with age a and gender g, the likelihood the user u selecting a music track is dependent on the music preferences of user age and gender as well as his/her personal music interest:

$$\begin{aligned} P(s|u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_t},\boldsymbol{\phi_v},\boldsymbol{\phi_s}) &= \lambda_u P(s|u,\boldsymbol{\theta_u},\boldsymbol{\phi_t},\boldsymbol{\phi_v},\boldsymbol{\phi_s}) \\ &+ \lambda_a P(s|a,\boldsymbol{\theta_a},\boldsymbol{\phi_t},\boldsymbol{\phi_v},\boldsymbol{\phi_s}) + \lambda_g P(s|g,\boldsymbol{\theta_g},\boldsymbol{\phi_t},\boldsymbol{\phi_v},\boldsymbol{\phi_s}) \end{aligned} \tag{4.1}$$

where $P(s|u,\boldsymbol{\theta_u},\boldsymbol{\phi_w},\boldsymbol{\phi_v},\boldsymbol{\phi_s})$ is the probability that song s is generated according to the personal music interest of user u, denoted as $\boldsymbol{\theta_u}$; $P(s|a,\boldsymbol{\theta_a},\boldsymbol{\phi_w},\boldsymbol{\phi_v},\boldsymbol{\phi_s})$ and $P(s|g,\boldsymbol{\theta_g},\boldsymbol{\phi_w},\boldsymbol{\phi_v},\boldsymbol{\phi_s})$ denote the probability that song s is generated according to the age music preference of a and gender music preference of g, denoted as $\boldsymbol{\theta_a}$ and $\boldsymbol{\theta_g}$ respectively. $\boldsymbol{\lambda} = \{\lambda_u, \lambda_a, \lambda_g : \lambda_u + \lambda_a + \lambda_g = 1\}$ is a categorial distribution, which controls the selection motivation of song s. That is, when selecting song s, it is possible that user u selects it according to his/her own music interests $\boldsymbol{\theta_u}$ with probability λ_u, or according to the age music preference $\boldsymbol{\theta_a}$ with probability λ_a, or according to the gender music preference $\boldsymbol{\theta_g}$ with probability λ_g. Note that $\boldsymbol{\lambda}$ is a group-dependent parameter, as users in different groups have different tendency to select music from different aspects. For example, from the training results, female users are more likely to select music tracks according to the general music preferences (namely, mainstreaming music) than male users. The generation process of UIA-MIT is shown in Algorithm 4 (Steps 5-32). Intuitively, UIA-MIT models user's music interests as the combination of the general music preferences according to certain user-specific information (age and gender here) and user's distinct music interests (affecting by user's personality, etc.). The general music preferences of certain user-specific information can be applied in music-related service. For example, in music retrieval and recommendation, more accurate results can be provided to new users when we know their age and/or gender based on the general age

and gender music preferences.

With the hyperparameters $\boldsymbol{\alpha} = \{\boldsymbol{\alpha_u}, \boldsymbol{\alpha_a}, \boldsymbol{\alpha_g}\}$, $\boldsymbol{\beta} = \{\boldsymbol{\beta_s}, \boldsymbol{\beta_w}, \boldsymbol{\beta_v}\}$ and $\boldsymbol{\gamma} = \{\gamma_u, \gamma_a, \gamma_g\}$, the joint distribution of the observed and hidden variables $\boldsymbol{s}$, $\boldsymbol{w}$, $\boldsymbol{v}$, $\boldsymbol{z}$, $\boldsymbol{y}$, and $\boldsymbol{\lambda}$ can be written as follows:

$$\begin{aligned} P(\boldsymbol{S}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{Z}, \boldsymbol{Y} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \int \cdots \int & P(\boldsymbol{S}, \boldsymbol{W}, \boldsymbol{V} | \boldsymbol{\phi_s}, \boldsymbol{\phi_t}, \boldsymbol{\phi_v}, \boldsymbol{Z}) \\ & P(\boldsymbol{Z} | \boldsymbol{y}, \boldsymbol{\theta_u}, \boldsymbol{\theta_a}, \boldsymbol{\theta_g}) P(\boldsymbol{\theta_u} | \boldsymbol{\alpha_u}) P(\boldsymbol{\theta_a} | \boldsymbol{\alpha_a}) P(\boldsymbol{\theta_g} | \boldsymbol{\alpha_g}) \\ & P(\boldsymbol{\phi_s} | \boldsymbol{\beta_s}) P(\boldsymbol{\phi_t} | \boldsymbol{\beta_t}) P(\boldsymbol{\phi_v} | \boldsymbol{\beta_v}) P(\boldsymbol{y} | \boldsymbol{\lambda}) P(\boldsymbol{\lambda} | \gamma_u, \gamma_a, \gamma_g) \\ & d\boldsymbol{\theta_u} d\boldsymbol{\theta_a} d\boldsymbol{\theta_g} d\boldsymbol{\phi_s} d\boldsymbol{\phi_t} d\boldsymbol{\phi_v} d\boldsymbol{\lambda} \end{aligned} \tag{4.2}$$

4.2.1.4 Model Inference

The estimation of the joint probability in Eq. 4.2 involves a set of parameters as shown. Among them, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are hyperparameters and pre-defined. The parameters to be estimated are (1) user music interest $\boldsymbol{\theta_u}$, (2) age music preference $\boldsymbol{\theta_a}$, (3) gender music preference $\boldsymbol{\theta_g}$, (4) topic-song distribution $\boldsymbol{\phi_s}$, (5) topic-text word distribution $\boldsymbol{\phi_w}$, (6) topic-audio word distribution $\boldsymbol{\phi_v}$. and (7) categorical distribution parameter $\boldsymbol{\lambda}$. Besides, in the generation process, we also need to assign the indicator vector $\boldsymbol{Y}$ and latent topic vector $\boldsymbol{Z}$ to the sequence of songs $\boldsymbol{S}$ in the corpus.

In our implementation, collapsed Gibbs sampling [46] is used to estimate the parameters in the topic models (MIT and UIA-MIT). Here, we describe how to inference the parameters in the UIA-MIT. The parameter inferences of the MIT model can be derived in a similar way. In the collapsed Gibbs sampling, the parameters $\{\boldsymbol{\theta_u}, \boldsymbol{\theta_a}, \boldsymbol{\theta_g}, \boldsymbol{\phi_s}, \boldsymbol{\phi_w}, \boldsymbol{\phi_v}\}$ and the categorical distribution parameter $\boldsymbol{\lambda}$ are estimated based on the results of a constructed Markov chain that converges to the posterior distribution on topic z. In the following, we present the sampling process to estimate the parameters $\{\boldsymbol{\theta_u}, \boldsymbol{\theta_a}, \boldsymbol{\theta_g}, \boldsymbol{\phi_s}, \boldsymbol{\phi_w}, \boldsymbol{\phi_v}, \boldsymbol{\lambda}\}$. The Collapsed Gibbs Sampling process of UIA-MIT is described in Algorithm 4.

Algorithm 4: Generative & Collapsed Gibbs Sampling Process for UIA-MIT

Input: A user music profile dataset D;
Number of topics: K;
Dirichlet hyperparameters: $\boldsymbol{\alpha_u}$, $\boldsymbol{\alpha_a}$, $\boldsymbol{\alpha_g}$, $\boldsymbol{\beta_s}$, $\boldsymbol{\beta_w}$, $\boldsymbol{\beta_v}$;
Categorical priors: γ_u, γ_a, γ_g

Output: Estimated parameters: $\hat{\boldsymbol{\theta}}_{\boldsymbol{u}}$, $\hat{\boldsymbol{\theta}}_{\boldsymbol{a}}$, $\hat{\boldsymbol{\theta}}_{\boldsymbol{g}}$, $\hat{\boldsymbol{\phi}}_{\boldsymbol{s}}$, $\hat{\boldsymbol{\phi}}_{\boldsymbol{w}}$, $\hat{\boldsymbol{\phi}}_{\boldsymbol{v}}$, $\hat{\lambda}_a$, $\hat{\lambda}_g$

Initialize $\boldsymbol{Z}$ and $\boldsymbol{Y}$ by assigning random values;
Count N_u^k, N_a^k, and N_g^k based on initialized $\boldsymbol{Z}$;
Count N_k^s, N_k^w, and N_k^v based on initialized $\boldsymbol{Z}$;
Count N_{y_0}, N_{y_1} and N_{y_2} based on initialized $\boldsymbol{Y}$;
for *each topic* $k = 1, ..., K$ **do**
    Draw $\boldsymbol{\phi_{k,s}} \sim Dir(\cdot|\boldsymbol{\beta_s})$;
    Draw $\boldsymbol{\phi_{k,w}} \sim Dir(\cdot|\boldsymbol{\beta_w})$;
    Draw $\boldsymbol{\phi_{k,v}} \sim Dir(\cdot|\boldsymbol{\beta_v})$;
for *each user* $u \in \mathcal{U}$ **do**
    Draw $\boldsymbol{\theta_u} \sim Dir(\cdot|\boldsymbol{\alpha_u})$;
for *each age range* $a \in \mathcal{A}$ **do**
    Draw $\boldsymbol{\theta_a} \sim Dir(\cdot|\boldsymbol{\alpha_a})$;
for *each gender* $g \in \mathcal{G}$ **do**
    Draw $\boldsymbol{\theta_g} \sim Dir(\cdot|\boldsymbol{\alpha_g})$;
repeat
    for *each user* $u \in \mathcal{U}$ *with age* $a \in \mathcal{A}$ *and gender* $g \in \mathcal{G}$ **do**
        for *each song* $s \in D_u$ **do**
            Toss a coin according to categorical distribution $y_s \sim Dir(\gamma_u, \gamma_a, \gamma_g)$;
            if $y_s == 0$ **then**
                Draw $z_s \sim Multi(\boldsymbol{\theta_u})$ according to the music interest of user u;
            if $y_s == 1$ **then**
                Draw $z_s \sim Multi(\boldsymbol{\theta_a})$ according to the music preference of age a;
            if $y_s == 2$ **then**
                Draw $z_s \sim Multi(\boldsymbol{\theta_g})$ according to the music preference of gender g;
            After the sampling of the topic $z_s = k$, draw song $s \sim Multi(\boldsymbol{\phi_{k,s}})$;
            for *each word* $w \in s_w$ **do**
                Draw $w \sim Multi(\boldsymbol{\phi_{k,w}})$;
            for *each audio word* $v \in s_v$ **do**
                Draw $v \sim Multi(\boldsymbol{\phi_{k,v}})$;
            Update N_{y_0}, N_{y_1} and N_{y_2} according to y_s;
            Update N_u^k, N_a^k, and N_g^k according to y_s and $z_s = k$;
            Update N_k^s, N_k^w, and N_k^v according to $z_s = k$;
until *convergence*;
Estimate model parameters $\{\hat{\boldsymbol{\theta}}_{\boldsymbol{u}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{a}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{g}}\}$, $\{\hat{\boldsymbol{\phi}}_{\boldsymbol{s}}, \hat{\boldsymbol{\phi}}_{\boldsymbol{w}}, \hat{\boldsymbol{\phi}}_{\boldsymbol{v}}\}$ and $\{\hat{\lambda}_a, \hat{\lambda}_g\}$ according to Eq. 4.7, Eq. 4.8 and Eq. 4.9, respectively

In the next, we show how to sample the topic z_s for song s (the steps 17-24 in Algorithm 4). Let $\boldsymbol{S}$ be a sequence of songs during the sampling process, $\boldsymbol{Z}$ and $\boldsymbol{Y}$ denote the set of topics and indicators corresponding to the song sequence. $\boldsymbol{W}$ and $\boldsymbol{V}$ denote the textual word and audio word sequence corresponding to $\boldsymbol{S}$. $\boldsymbol{S}_{\neg i}$ denotes $\boldsymbol{S}$ excluding song s_i, and $\boldsymbol{W}_{\neg i}$ denotes $\boldsymbol{W}$ excluding words in w_i. Similar notations are used for other variables. Let s_i in the profiles of a user in user age group a and gender group g, for $\boldsymbol{S} = \{s_i, \boldsymbol{S}_{\neg i}\}$, $\boldsymbol{W} = \{\boldsymbol{w}_i, \boldsymbol{W}_{\neg i}\}$, $\boldsymbol{V} = \{\boldsymbol{v}_i, \boldsymbol{V}_{\neg i}\}$, $\boldsymbol{Z} = \{z_i, \boldsymbol{Z}_{\neg i}\}$, and $\boldsymbol{Y} = \{y_i, \boldsymbol{Y}_{\neg i}\}$, we show how to jointly sample y_i and z_i of a song s_i and its words ($\boldsymbol{w}$ and $\boldsymbol{v}$). y_i and z_i are needed to be sampled jointly, because the value of y_i decides sampling z_i from $\theta_u(y_i = 0)$, $\theta_a(y_i = 1)$ or $\theta_g(y_i = 2)$ as shown in Fig. 4.3. The joint probability of sampling $z_i = k$ and $y_i = 0$ is:

$$P(z_i = k, y_i = 0 | \boldsymbol{Z}_{\neg i}, \boldsymbol{Y}_{\neg i}, \boldsymbol{S}, \boldsymbol{W}, \boldsymbol{V}, \cdot) \propto (\gamma_u + N_{y_0,\neg i}) \cdot \frac{N^k_{u,\neg i} + \alpha_u}{\sum_{k=1}^{K}(N^k_{u,\neg i} + \alpha_u)} \cdot P_{com} \tag{4.3}$$

$$\begin{aligned} P_{com} =& \frac{\beta_s + N^s_{k,\neg i}}{\sum_{s=1}^{S}(\beta_s + N^s_{k,\neg i})} \cdot \frac{\prod_{t \in \boldsymbol{w_{s_i}}}(\beta_t + N^{s,t}_k)!}{\prod_{t \in \boldsymbol{w_{s_i}}}(\beta_t + N^{s,t}_k - N^t_s)!} \frac{(\sum_{t=1}^{T}(\beta_t + N^{s,t}_k - N^t_s))!}{(\sum_{t=1}^{T}(\beta_t + N^{s,t}_k))!} \\ & \cdot \frac{\prod_{v \in \boldsymbol{v_{s_i}}}(\beta_v + N^{s,v}_k)!}{\prod_{v \in \boldsymbol{v_{s_i}}}(\beta_v + N^{s,v}_k - N^v_s)!} \frac{(\sum_{v=1}^{V}(\beta_v + N^{s,v}_k - N^v_s))!}{(\sum_{v=1}^{V}(\beta_v + N^{s,v}_k)!} \end{aligned} \tag{4.4}$$

where N^k_u denotes the number of times that topic k is sampled from the topic distribution $\boldsymbol{\theta_u}$, denoting the music interest of user u. N_{y_0} is the number of times that the topics drawn from user's music interest $\boldsymbol{\theta_u}$. N^s_k is the number of times that song s is sampled from topic k. The number $N^{\cdot}_{\cdot,\neg i}$ with $N_{\neg i}$ denote a quantity, excluding the current instance. $N^{s,w}_k$ and $N^{s,v}_k$ denote the number of times of text word w and audio word v assigned to topic k because of their occurrence times in song s (since s is associated to topic k). $\boldsymbol{s_{i,w}}$ denotes the text word sequence in song s; and $\boldsymbol{s_{i,v}}$ denotes the audio word sequence in song v. N^w_s is the occurrence time of text word w appearing in song s; and N^v_s is the occurrence time of audio word v in song s. Notice that the exclusion of s_i from $\boldsymbol{S}$ will cause the exclusion of $\boldsymbol{s_{i,w}}$ from $\boldsymbol{W}$ and $\boldsymbol{s_{i,v}}$ from $\boldsymbol{V}$. Consequently, the

words $w \in \boldsymbol{s_{i,w}}$ and $v \in \boldsymbol{s_{i,v}}$ will be excluded from the topic $z_i = k$ for multiple times (N_s^w and N_s^v, respectively; $N_{k,\neg i}^{s,w} = N_k^{s,w} - N_s^w$ and $N_{k,\neg i}^{s,v} = N_k^{s,v} - N_s^v$). The influence of the exclusion of s on the topic distribution is represented in Eq. 4.4. The three items separated by a dot $\cdot$ on the right hand side of the equation denote the caused topic distribution changes corresponding to the exclusion of song s_i, song's text words $\boldsymbol{s_{i,w}}$ and song's audio words $\boldsymbol{s_{i,v}}$, respectively.

Similarly, the joint probabilities of sampling $z_i = k$ for the cases of $y_i = 1$ and $y_i = 2$ are:

$$P(z_i = k, y_i = 1 | \boldsymbol{Z}_{\neg i}, \boldsymbol{Y}_{\neg i}, \boldsymbol{S}, \boldsymbol{W}, \boldsymbol{V}, \cdot) \propto (\gamma_a + N_{y_1,\neg i}) \cdot \frac{N_{a,\neg i}^k + \alpha_a}{\sum_{k=1}^K (N_{a,\neg i}^k + \alpha_a)} \cdot P_{com} \quad (4.5)$$

$$P(z_i = k, y_i = 2 | \boldsymbol{Z}_{\neg i}, \boldsymbol{Y}_{\neg i}, \boldsymbol{S}, \boldsymbol{W}, \boldsymbol{V}, \cdot) \propto (\gamma_g + N_{y_2,\neg i}) \cdot \frac{N_{g,\neg i}^k + \alpha_g}{\sum_{k=1}^K (N_{g,\neg i}^k + \alpha_g)} \cdot P_{com} \quad (4.6)$$

where N_a^k and N_g^k denote the number of times that topic k is sampled from the topic distribution of the music preference of age a and the music preference of gender g, respectively. N_{y_1} and N_{y_2} are the number of times the topics are drawn from age music preference $\boldsymbol{\theta_a}$ and gender music preferences $\boldsymbol{\theta_g}$, respectively.

After a sufficient number of sampling iterations, based on the state of the Markov chain Y and Z, the parameters can be estimated:

$$\theta_{u,k} = \frac{\alpha_u + N_u^k}{\sum_{k'} (\alpha_u + N_u^k)}, \quad \theta_{a,k} = \frac{\alpha_a + N_a^k}{\sum_{k'} (\alpha_a + N_a^k)}, \quad \theta_{g,k} = \frac{\alpha_g + N_g^k}{\sum_{k'} (\alpha_g + N_g^k)} \quad (4.7)$$

$$\phi_{k,s} = \frac{\beta_s + N_k^s}{\sum_{s'} (\beta_s + N_k^s)}, \quad \phi_{k,t} = \frac{\beta_t + N_k^t}{\sum_{t'} (\beta_t + N_k^t)}, \quad \phi_{k,v} = \frac{\beta_v + N_k^v}{\sum_{v'} (\beta_v + N_k^v)} \quad (4.8)$$

$$\lambda_u = \frac{\gamma_u + N_{y_0}}{\gamma_a + \gamma_u + \gamma_g + N_{y_0} + N_{y_1} + N_{y_2}}, \quad \lambda_a = \frac{\gamma_a + N_{y_1}}{\gamma_a + \gamma_u + \gamma_g + N_{y_0} + N_{y_1} + N_{y_2}} \quad (4.9)$$

N_k^w and N_k^v are the number of times text word w and audio word v sampled from topic k in the corpus, respectively. $\hat{\lambda}_g$ can be obtained by $1 - \hat{\lambda}_u - \hat{\lambda}_a$.

4.2.2 Music Retrieval based on MIT and UIA-MIT

In this section, we show how to apply the proposed topic models for exploiting the age and gender information in music retrieval. Notice that in MIT and UIA-MIT, music concepts (tags, or text word w) and songs are associated (by latent topics) in the latent music interest space, which is discovered based on the music preferences of users. Thus, the captured associations can be used to estimate the relevance between concepts and songs, which reflects user's music preferences on the songs with respect to the concepts. Thus, the MIT and UIA-MIT can be used for semantic-based music retrieval. And the retrieved results consider the collaborative preferences of general users with respect to the query.

Given the query q, for each song, a probability $P(s|q)$ can be estimated by both MIT and UIA-MIT models. In music retrieval or re-ranking, the candidate songs are ranked in the descending order of $P(s|q)$ and the top results are returned to the user. Specifically, for a query $q = \{w_1, w_2.., w_n\}$, the conditional probability $P(s|q)$ is estimated based on the estimated parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta_u}, \boldsymbol{\theta_a}, \boldsymbol{\theta_g}\}$ and $\boldsymbol{\Phi} = \{\boldsymbol{\phi_s}, \boldsymbol{\phi_t}\}$.

$$p(s|q, \boldsymbol{\Theta}, \boldsymbol{\Phi}) \propto P(s, q|\boldsymbol{\Theta}, \boldsymbol{\Phi}) = \prod_{i=1}^{n} P(t_i|s, \boldsymbol{\Theta}, \boldsymbol{\phi_t}) P(s|\boldsymbol{\Theta}, \boldsymbol{\phi_s}) \tag{4.10}$$

The words in the query q are assumed to be independent in the above equation. In MIT model, $\boldsymbol{\Theta} = \{\boldsymbol{\theta_u}\}$; and in UIA-MIT model, $\boldsymbol{\Theta} = \{\boldsymbol{\theta_u}, \boldsymbol{\theta_a}, \boldsymbol{\theta_g}\}$. The UIA-MIT model can incorporate age and gender information in retrieval. Fig. 4.4 shows the retrieval procedure using the UIA-MIT model.

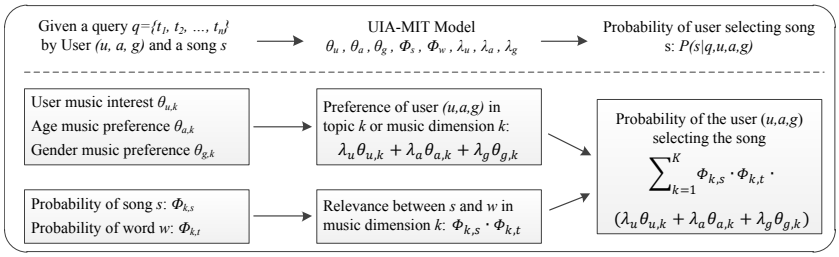


Figure 4.4: The scheme of using UIA-MIT in text-based music retrieval.

4.2.2.1 Music Retrieval based on MIT

According to Eq. 4.10, the conditional probability $P(s|q)$ in the MIT model becomes:

$$P(s|q,\boldsymbol{\theta},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) \propto P(s,q|\boldsymbol{\theta},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) = \prod_{i=1}^{n} P(t_i|s,\boldsymbol{\theta},\boldsymbol{\phi_t})P(s|\boldsymbol{\theta},\boldsymbol{\phi_s}) \tag{4.11}$$

where $P(s|\theta,\phi_s)$ is computed as:

$$P(s|\boldsymbol{\theta},\boldsymbol{\phi_s}) = \sum_{k=1}^{K} P(s|z=k,\boldsymbol{\theta},\boldsymbol{\phi_s})P(z=k|\boldsymbol{\theta}) = \sum_{k=1}^{K} \phi_{k,s}\cdot\theta_k \tag{4.12}$$

According to Bayes rule and the graphical representation of the MIT model:

$$\begin{aligned} P(t_i|s,\boldsymbol{\theta},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) &= \sum_{k=1}^{K} P(t_i|z=k,\boldsymbol{\phi_t})P(z=k|s,\boldsymbol{\theta},\boldsymbol{\phi_s}) \\ &= \sum_{k=1}^{K} P(t_i|z=k,\boldsymbol{\phi_t})\cdot\frac{P(s|z=k,\boldsymbol{\theta},\boldsymbol{\phi_s})P(z=k|\boldsymbol{\theta})}{P(s|\boldsymbol{\theta},\boldsymbol{\phi_s})} \\ &= \sum_{k=1}^{K} \phi_{k,t_i}\cdot\frac{\phi_{k,s}\cdot\theta_k}{\sum_{k=1}^{K}\phi_{k,s}\cdot\theta_k} \end{aligned} \tag{4.13}$$

Based on Eq. 4.12 and Eq. 4.13, Eq. 4.11 becomes:

$$P(s|q,\boldsymbol{\theta},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) \propto \prod_{i=1}^{n}\sum_{k=1}^{K}\phi_{k,t_i}\cdot\phi_{k,s}\cdot\theta_k \tag{4.14}$$

For the user u whose music profile is known, $\boldsymbol{\theta}$ in the above equations is his/her music preference $\boldsymbol{\theta_u}$ obtained in the MIT, and $\boldsymbol{\theta_k}$ in the equation is $\boldsymbol{\theta_{u,k}}$. While in the case of without any prior knowledge about the user, $P(z=k|\boldsymbol{\theta})$ is set to the normalized probability of $P(z=k|\boldsymbol{\theta_u}) = \boldsymbol{\theta_{u,k}}$ over all users in the corpus.

4.2.2.2 Music Retrieval based on UIA-MIT

In many scenarios, it is possible to know some information about the user, such as age and gender associated with the users' accounts registered in music related applications or services. In such cases, the age and gender information can be utilized to improve the search results. In the UIA-MIT model, with the age and gender information, Eq. 4.11 - Eq. 4.13 becomes:

$$\begin{aligned} &P(s|q,u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) \\ &\quad = \prod_{i=1}^{n} P(t_i|s,u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_t}) \cdot P(s|u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_s}) \end{aligned} \tag{4.15}$$

$$\begin{aligned} P(s|u,a,g,\cdot) &= \sum_{k=1}^{K} P(s|z=k,\boldsymbol{\phi_s}) P(z=k|u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g}) \\ &= \sum_{k=1}^{K} P(s|z=k,\boldsymbol{\phi_s})(\lambda_u P(z=k|u,\boldsymbol{\theta_u}) + \lambda_a P(z=k|a,\boldsymbol{\theta_a}) + \lambda_g P(z=k|g,\boldsymbol{\theta_g})) \\ &= \sum_{k=1}^{K} \phi_{k,s} \cdot (\lambda_u \theta_{u,k} + \lambda_a \theta_{a,k} + \lambda_g \theta_{g,k}) \end{aligned} \tag{4.16}$$

$$\begin{aligned} P(t_i|s,\cdot) &= \sum_{k=1}^{K} P(t_i|z=k,\boldsymbol{\phi_t}) \frac{P(s,z_j|u,a,g,\cdot)}{P(s|u,a,g,\cdot)} \\ &= \sum_{k=1}^{K} \phi_{k,t_i} \cdot \frac{\phi_{k,s} \cdot (\lambda_u \theta_{u,k} + \lambda_a \theta_{a,k} + \lambda_g \theta_{g,k})}{\sum_{k=1}^{K} \phi_{k,s} \cdot (\lambda_u \theta_{u,k} + \lambda_a \theta_{a,k} + \lambda_g \theta_{g,k})} \end{aligned} \tag{4.17}$$

Based on Eq. 4.16 and Eq. 4.17, Eq. 4.15 becomes

$$P(s|q,u,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \phi_{k,t_i} \cdot \phi_{k,s} \cdot (\lambda_u \theta_{u,k} + \lambda_a \theta_{a,k} + \lambda_g \theta_{g,k}) \tag{4.18}$$

When the user $\boldsymbol{\theta_u}$ is known, Eq. 4.18 can be used for personalized music search. In the case of that $\boldsymbol{\theta_u}$ of the user is unavailable and his/her age and gender are known, λ_a and λ_g are normalized to keep $\lambda_a + \lambda_g = 1$, and the following equation is used for retrieval:

$$P(s|q,a,g,\boldsymbol{\theta_u},\boldsymbol{\theta_a},\boldsymbol{\theta_g},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \phi_{k,t_i} \cdot \phi_{k,s} \cdot (\lambda_a \theta_{a,k} + \lambda_g \theta_{g,k}) \tag{4.19}$$

If either age or gender information is available, only the corresponding music preferences will be used (namely, set $\lambda_a = 1$ or $\lambda_g = 1$ in the equation). Intuitively, $\phi_{k,w_i} \cdot \phi_{k,s}$ evaluates the similarity of the song s with respect to query q in the music dimension k in the music interest space, and $\lambda_a \theta_{a,k} + \lambda_g \theta_{g,k}$ estimates the music preferences with respect to age range a and gender g in the music dimension k. Thus it can be seen as that the model re-weights the original query in different music dimensions based on user's age and gender information. Notice that the UIA-MIT model can be used in personalized music search, while it suffers from the cold-start problem for new users. As it is usually easier to know user's age and gender information, such as the account registration with user information in music services like Last.fm, the exploitation of age and gender (i.e., Eq. 4.19) can alleviate the cold-start problem in personalized music retrieval.

4.2.3 Model Extendability

The UIA-MIT model is easily extended to incorporate other user information, such as country, culture, etc. For example, when m factors are considered, let $\boldsymbol{f} = \{f_1, f_2, \cdots, f_m\}$ denote the set of considered factors, based on the extended model, the likelihood of a user u selecting a song s (namely Eq. 4.1) becomes:

$$P(s|u, \boldsymbol{f}, \boldsymbol{\Theta}, \boldsymbol{\Phi}) = \sum_{i=1}^{m} \lambda_i P(s|u, f_i, \boldsymbol{\theta_i}, \boldsymbol{\Phi}) \tag{4.20}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \cdots, \boldsymbol{\theta_m}\}$, $\boldsymbol{\Phi} = \{\boldsymbol{\phi_s}, \boldsymbol{\phi_t}, \boldsymbol{\phi_v}\}$, and $\boldsymbol{\theta_i}$ denotes the music preference of factor f_i (e.g., age, gender, or country). λ_i is the probability of selecting the song according to the music preference of factor f_i. Correspondingly, in the retrieval, Eq. 4.19 becomes:

$$P(s|q, \boldsymbol{f}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \phi_{k,t_i} \cdot \phi_{k,s} \cdot \sum_{j=1}^{m} \lambda_j \cdot \theta_{j,k} \tag{4.21}$$

4.3 Experimental Setup

In previous sections, we observed the influence of age and gender information on music preferences and proposed a retrieval method based on the UIA-MIT model, which can incorporate age and/or gender information in music retrieval. In the next, we would like to validate the effectiveness of exploiting age and gender information in music retrieval. We design an offline experimental study on text-based music retrieval to compare the proposed method with several competitors which are currently popular and the state-of-the-art text-based music retrieval methods. In summary, the experiments answer the following research questions:

RQ1 Whether the UIA-MIT model can capture the music preferences of different ages and genders? The results directly relate to the performance of the UIA-MIT based retrieval methods. (Sect. 4.4.2.1)

RQ2 Comparing to other text-based music retrieval methods, how well can UIA-MIT based retrieval methods perform on retrieval with the use of age and/or gender information? (Sect. 4.4.2.1)

RQ3 Whether the utilization of user-specific information (i.e., age and gender information in our study) for re-ranking can improve the music search accuracy? If so, how much can be improved using age and/or gender information? (Sect. 4.4.2.2 and Sect. 4.4.3)

RQ4 Whether the UIA-MIT model can be extended to capture the music preference of other user information (e.g., country) for retrieval? (Sect. 4.4.3)

RQ1 checks whether the proposed UIA-MIT model can capture the influence of age and gender on music interests by examining the topic distributions of different ages and genders. **RQ2** is to study the effectiveness of the retrieval methods based on the UIA-MIT model using age and/or gender information

in ad-hoc search. **RQ3** investigates whether our proposed methods can effectively use the age and/or gender information to refine the search results. We use the estimated probability $P(s|q)$ based on the UIA-MIT model to re-rank the results of other text-based retrieval methods to check the performance improvement. Besides, we also explore the effectiveness of user-specific information in retrieval by examining the performance of a re-ranking method which uses the age and gender information in a heuristic way (see Music Popularity Based Re-ranking in Sect. 4.3.2.1). **RQ4** is to demonstrate the extendability of the UIA-MIT model on considering other user information (i.e., country) in the model.

4.3.1 Datasets

To evaluate the search accuracy of retrieval systems with respect to query users, a great challenge is how to get the ground truth of the test queries with respect to corresponding query users. In our retrieval task, given a query q of a user in a group, a relevant song should be not only relevant to the query but also loved by the users in this group. We develop two test collections by crawling user information from Last.fm. Hundreds of queries and corresponding ground truth are generated for each test collection.

User Profile Dataset. To learn the music preferences via UIA-MIT, we construct a dataset with users' demographic information and their loved music tracks from Last.fm. The dataset is collected in the following procedures. 160 recent active users were randomly selected from Last.fm[3]. Then the friends of these users and the friends of their friends were collected with their demographic information, including age, gender, country. In total, 90,036 users were collected. The loved tracks of these users were collected using Last.fm public API "User.getLovedTracks".

[3]Accessed http://www.last.fm/community/users/active on Mar 3, 2015.

Test Collection 1 (TC1). To judge the relevance of songs with respect to queries, it is necessary to label the songs in the dataset with used query concepts. CAL10K [150] is a labeled song collection. The annotations are used as ground truth in previous text-based music retrieval research [101]. This dataset contains 10,870 songs from 4,597 different artists. The label vocabulary is composed of 137 "genre" tags and 416 "acoustic" tags. The number of tags of songs varies from 2 to 25 tags. The song tags are mined from the Pandora Web sites. The annotations in Pandora are contributed by music experts and are considered highly objective [150].

In order to train UIA-MIT, the user profile dataset was processed to only keep the users, whose age and gender information is available. As the number of users with age under 16 or above 54 years old is small, this study focuses on studying the influence of ages between 16 to 54 years old. After filtering, there are 45,334 users left. For user's loved tracks, we attempted to download their audio streams from 7digital[4]. 7digital provides 30s audio stream for songs. Based on the successfully downloaded tracks, we removed the users with less than 10 loved songs and songs loved by less than 10 users. Finally, there are 29,412 users in the use set and 15,323 songs in the song set. The social tags of these songs are collected from Last.fm using API "Track.getTopTags". The social tags of songs from Last.fm are used in the topic model training and the tag-based music retrieval method (see the TAG method in Section 4.3.2.1). In the song set, there are 2,839 songs contained in the CAL10K dataset, which is used as the retrieval collection. The user set contains 15,826 males and 13,586 females. In our implementation, we categorized the users into 7 age groups. Thus, there are 14 user groups (7 age groups $\times$ 2 gender groups). The numbers of users in different age groups are shown in Table 4.1.

Test Collection 2 (TC2). TC1 could be used to test the performance of the proposed retrieval method by leveraging user's age and gender informa-

[4]https://www.7digital.com/

Table 4.1: Number of users in each age group in Test Collection 1.

Group	0	1	2	3	4	5	6
Age	16-20	21-25	26-30	31-35	36-40	41-45	46-54
# Users	9003	12,820	4941	1482	595	324	247

tion. However, the dataset size for retrieval is relatively small, with only 2,839 songs, due to the limited size of the well-labeled songs. As social tags have been used for text-based music retrieval, in TC2, we use social tags as annotations in relevance judgment. Besides, with the increasing size of the retrieval dataset, the TC2 also could be used to validate the extendability of UIA-MIT to other user information, e.g., country.[5] In the User Profile Dataset, there are 26,468 users, who provide age, gender, and country information. The users are from 179 different countries. With the age, gender, and country information, we categorize users into groups based on {age, gender, country}, e.g., 16-20_male_US. Since the number of users in groups affects the performance of UIA-MIT on capturing the music preference of users in different groups, to ensure that each group has a relatively large number of users, three age groups {16-20, 21-25, 26-30} and five countries {Brazil, US, UK, Poland, Russian} are used in experiments. In total, there are 30 groups, and the numbers of users in these groups are shown in Table 4.2. Based on users in these groups, we further removed the users with less than 10 loved songs and songs loved by less than 10 users. Finally, there are 14,715 users and 1, 0197 songs. All the songs are used in retrieval.

Query Set. The best choice of query set is to collect user generated queries in real applications. The problem is that it is possible that our test collections do not contain enough relevant songs for the collected queries. If there are many queries, which only have several relevant songs or even none relevant

[5]Because there are only 2,836 songs in the retrieval dataset, it is hard to generate lots of queries, which have enough relevant songs in each user group with the constraint of age, gender, and country.

Table 4.2: Number of users in different groups in Test Collection 2.

Country	Male			Female		
	16-20	21-25	26-30	16-20	21-25	26-30
Brazil	1605	1320	253	1665	809	156
Poland	179	270	83	472	401	98
Russia	83	269	109	184	314	92
UK	186	416	223	198	364	120
US	539	1360	684	457	1273	533

songs in a retrieval dataset, then the dataset cannot be used to provide high-quality performance evaluation over different retrieval methods. An alternative method is to select the frequent tags as queries, as a main functionality of social tags is used for retrieval, such as the social tags in Flickr[6] and Last.fm. In experiments, we use a combination of k distinct terms as queries. Following the methodology in [101, 151], queries composed by $k = \{1, 2, 3\}$ terms are used. The method described in [101] is used to construct the query set. In TC1, all the terms in CAL10K dataset are treated as 1-term query candidates. And for 2-term and 3-term queries, all the term combinations are considered as candidates. Then, we filtered the query candidates by only keeping the queries with at least 10 relevant songs in the ground truth in all user groups (14 user groups). Finally, there are 33 1-term queries, 122 2-term queries and 542 3-tag query in total. In TC2, social tags are used to generate the queries. We first filtered the tags which appear less than 10 times in the dataset, and then removed the tags which express personal interests in the song, such as "favorite", "great", "favor", "excellent", etc. Then, we tokenized the tags of each song into terms. Similarly, all the terms are treated as candidates. For 2-term and 3-term queries, all the term combinations which appear in a song, are treated as candidates. In the next step, we filtered the query candidates by only keeping the queries with at least 10 relevant songs in the ground truth in

[6]https://www.flickr.com/

all user groups (30 user groups). This leads to 79 1-term queries, 1691 2-term queries, and 12,584 3-term queries. For the 3-term queries, we retain a random sample of 3,000 queries as [101], with the assumption that they are generally enough to evaluate the task. Table 4.3 summarizes the number of queries in TC1 and TC2. Notice that the queries for each group are the same. Table 4.4 gives some query examples for each type.

Table 4.3: Number of queries in TC1 and TC2.

Test Collection	# 1-Term Query	# 2-Term Query	# 3-Term Query
TC1	33	122	542
TC2	79	1691	3000

Table 4.4: Few examples for each type of queries.

1-Term Query	2-Term Query	3-Term Query
aggressive	aggressive, guitar	aggressive, angry, guitar
angry	aggressive, rock	angry, guitar, rock
breathy	bass, tonality	drums, angry, guitar
country	blues, guitar	guitar, aggressive, angry
danceable	country, guitar	guitar, pop, romantic
electronica	danceable, harmonies	mellow, folk, tonality
guitar	drums, guitar	mellow, guitar, rock
mellow	emotional, romantic	organ, piano, tonality
piano	mellow, pop	piano, guitar, rock
romantic	rock, tempo	rock, aggressive, angry

Ground Truth. As the query is subject to each user group, a relevant song with respect to a query should (1) contains all the query terms in the annotations (in the CAL10K dataset for TC1) or social tags (for TC2); and (2) be loved by at least 10 users in this user group. Based on the criteria, the relevant songs in the retrieval datasets of TC1 and TC2 are labeled. For each query, the numbers of relevant songs in different groups are different. Notice that we used two-fold cross-validation in experiments (as shown in Sect. 4.3.2).

4.3.2 Experimental Configurations

In experiments, we split the users into two folds and use two-fold cross-validation: one fold (users with their loved tracks) is used to training the models (MIT and UIA-MIT), and the other fold is used to create the query set and generate the corresponding ground truth. The dataset is split in the way to guarantee each group has approximately equal number of users in the two folds. In the result presentation, the presented results are the average performance over the two folds. In the MIT and UIA-MIT models, the generation of the three types of documents are presented below.

User-Song Document For each user, a user-song document is generated based on his/her loved songs. The document is comprised by the concatenation of the songs loved by the users.

Song-Text Word Document. The social tags of songs from Last.fm (collected using API "Track.getTopTags") are used to form the text documents for songs. In our implementation, the tags that appeared less than 10 songs are filtered, and then the remaining tags of a song are concatenated together and tokenized with a standard stop-list to form the text document for the song.

Song-Audio Word Document. For each song, the 30 seconds audio track downloaded from 7digital is used to generate the "bag-of-audio-word" document. In our implementation, an audio word is an acoustic feature vector computed on half-overlapping windows of 23 milliseconds. MFCCs and Chroma features are used to generate the audio words. MFCCs are the most popular timbre feature used in music retrieval. For each frame, a 13-dimensional MFCC vector with its first and second instantaneous derivatives are extracted, achieving a final 39-dimensional MFFCs feature. The Chroma feature is a 12-dimensional, real-valued vector that approximates an audio signal's strength at each music note. Chroma features are considered because they are invariant to types of distortions that affect timbre and somewhat invariant

to certain differences between renditions and arrangements, which are useful in detecting songs. Before extraction, each song is converted to a standard mono-channel and 22,050 Hz sampling rate WAV format, a common practice in music information retrieval. openSMILE [43] is used to extract both features. After feature extraction, K-means clustering method is then used to group frames into clusters based on their feature vectors. The cluster centers are used as audio words. Replacing each frame with the nearest audio word, the music track is represented as a sequence of audio words. In experiments, we empirically set the number of audio words to 4096.

4.3.2.1 Competitors and Evaluation metrics

To explore the effectiveness of using age and/or gender information retrieval by our proposed retrieval methods, variants of our methods using age and/or gender information are tested in the experiments and compared with currently popular and the state-of-the-art text-based retrieval methods. To demonstrate the superiority of our methods on utilizing the age and gender information, we also compare our methods with the ones which use age and gender information. In the following, we first present the variants of our methods, and then other retrieval methods.

Proposed Method and Variants. Several variants are used to investigate the improvements by using age and gender information individually and together. These variants include:

- MIT: this is the retrieval method based on the MIT model, which has not considered any user-specific information. The method is described in Sect. 4.2.2.1 (Eq. 4.14);
- A-MIT: this method simulates the search scenarios when only age information is available for the UIA-MIT based retrieval method, namely, set $\lambda_a = 1$ in Eq. 4.19 ;

- G-MIT: this method simulates the search scenarios when only gender information is available for the UIA-MIT based retrieval method, namely, set $\lambda_g = 1$ in Eq. 4.19;
- C-MIT: in this method, the UIA-MIT model considers age, gender, and country information simultaneously; and C-MIT simulates the search scenarios when only country information is available for the UIA-MIT based retrieval method (only used in Test Collection 2);
- AG-MIT: this method simulates the search scenarios when both age and gender are available for the UIA-MIT based retrieval method (Eq. 4.19).
- AGC-MIT: in this method, the UIA-MIT model considers age, gender, and country information simultaneously; and AGC-MIT simulates the search scenarios when user's age, gender and country information are available (only used in Test Collection 2).

Methods without Using Age/Gender Information. We considered the following text-based music retrieval methods:

- **Tag-based method (TAG)**: In this method, the social tags of each song in Last.fm are used as the text description for retrieval. Tags are first tokenized with a standard stop-list, and then a conventional document-term matrix is created by tabulating the number of occurrences of each word in tags. The standard tf-idf weighting scheme is used to compute the similarity between query and songs with the standard cosine distance in the Vector Space Model [129].
- **Weighted Linear Combination (WLC)**: Similar to the WLC described in [101], the first result returned by TAG is used as the seed for a content-based music retrieval (CBMR) method. Then the score of the TAG method and CBMR method are linearly combined together to

generate the final search results. The weighted linear combination can be described as follows,

$$Sim(q, s) = w \cdot TAG(q, s) + (1 - w) \cdot CBMR(q, s) \tag{4.22}$$

where $TAG(q, s)$ is the similarity score obtained by TAG method and $CBMR(q, s)$ is the similarity score obtained by CBMR method. The CBMR method described in [127] is used in our experiments. Specifically, the "audio words" are treated as text terms, and then standard VSM method is used to retrieve the music given the seed song as the query. The combination weights are tuned for obtaining the highest MAP in experiments.

- **Post-Hoc Audio-based Reranking (PAR)** [70]: It is a method to incorporate audio similarity into an already existing ranking. In our experiments, the results of tag-based method (TAG) are used as the initial ranking list. PAR computes a new score for each song s by considering the original rank of s, the original ranks of all the songs having s in their acoustic neighborhood, and the rank of s in all these neighborhoods. For more details, please refer to the paper [70]. In the implementation, we followed the details reported in the referred paper.

Methods using Age and Gender Information. To the best of our knowledge, we have not found any music search methods using user-specific information in the retrieval algorithm. Thus, we compare our methods with the following two heuristic methods on utilizing age and gender information. Notice that if these methods can also improve the search accuracy, it further demonstrates the importance of considering user-specific information in music retrieval.

- **Music Popularity Based Re-ranking (MPR)**: As users in different

ages and genders prefer different songs with respect to a query, such as *rock*, a heuristic method is to re-rank the search results (returned by other search methods) according to the songs' popularity in different age and gender group. Thus, we implemented a popularity-based re-ranking method to re-rank the top 100 songs returned by other retrieval methods (such as the above ones) according to the songs' popularity score in the targeted user's age and gender group. Specifically, given a query of a user u with age a and gender g, and a returned song list $s \in L(q, a, g)$, popularity-based re-ranking method is to re-rank $s \in L(q, a, g)$ to obtain a new song list L' in the descending order of song's popularity $POP(s)$ in the user's age and gender group (a, g). The popularity score is computed as:

$$POP(s) = \frac{N(s, a, g)}{N(a, g)} \tag{4.23}$$

where $N(s, a, g)$ is the number of users in group (a, g) loving song s, and $N(a, g)$ is the total number of the users in group (a, g).

- **Group User Music Representation (GUMR)**: In this method, we create music representations of users in group (a, g) according to their music preferences. Specifically, for each group, we aggregate the social tags of songs loved by the users in this group to form a document for the group. The same procedure in Tag-based method (TAG) is used to process the document, and then the similarity distance between a group document to each song is computed using cosine distance based on the standard tf-idf weighting scheme. Then for a given query q of user u in age a and gender g, the similarity score of a song s with respect to the query is computed as:

$$Sim(q, s, a, g) = w \cdot TAG(q, s) + (1 - w) \cdot Sim(s, a, g) \tag{4.24}$$

in which $TAG(q, s)$ is the cosine similarity between the song s and q using TAG method, and $Sim(s, a, g)$ is the similarity between user group preference and the song s. Thus, this method considers both the relevance of query q with respect to the song and the music preference of users in a specific age and gender. The combination weight is tuned in experiments.

In the above methods, PAR and MPR are re-ranking methods and thus compare with the proposed methods and variants in the re-ranking task (Sect. 4.4.3). Other methods are compared in the ad-hoc search (Sect. 4.4.2).

In experiments, we focus on the evaluation of the search accuracy and use standard information retrieval metrics, which are used in previous music retrieval tasks [101]. Specifically, the following three metrics are used: Precision at k (P@k), Mean Reciprocal Rank (MRR), Mean Average Precision (MAP). Details about the metrics are in Appendix A.

4.3.2.2 Parameter Setting

In implementation, the hyperparameters in the topic model are turned in a wide range: in the UIA-MIT model, without prior knowledge about the topic distributions of users in different ages and genders, we set $\boldsymbol{\alpha_u}$, $\boldsymbol{\alpha_a}$ and $\boldsymbol{\alpha_g}$ to be symmetric. For simplicity, we set them to be the same and tune them in the range of $\alpha = \alpha_u = \alpha_a = \alpha_g \in \{0.01, 0.05, 0.1, 1.0, 5.0\}$. Similarly, β_t, β_v and β_s are also set to be symmetric and the same: $\beta = \beta_t = \beta_v = \beta_s \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.25\}$. The values of γ_u, γ_a and γ_g bias the tendency of choosing music according to user's personal, age or gender music preferences. We would like the tendency to be learned from the data, thus γ_u, γ_a, γ_g are all set to 1. In Gibbs sampling for the training of topic models, 100 sampling iterations were run as burn-in iterations and then 50 sampling iterations with a gap of 10 were taken to obtain the final results. For the WLC and GUMR

methods, the weight w (in Eq. 4.22 and Eq. 4.24) is turned in the range [0.05, 1] with an interval 0.05. In the result presentation in Sect. 4.4, the reported results are based on the parameters with the best results.

4.4 Experimental Results

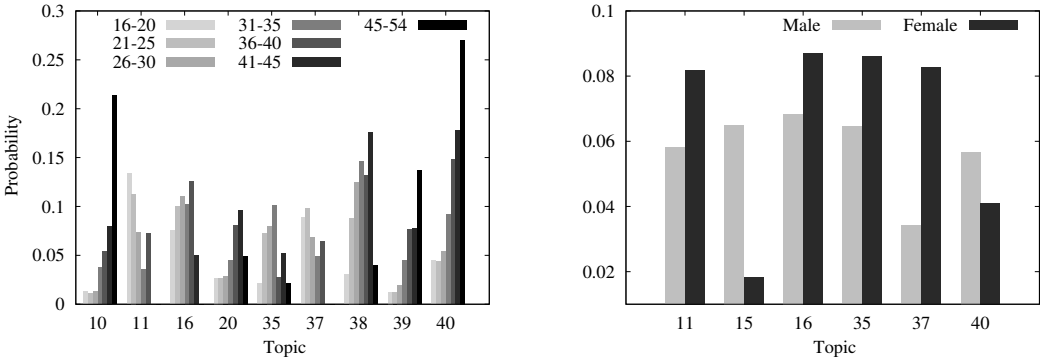
In Sect. 4.4.1, we report a qualitative study on the proposed UIA-MIT model by presenting the music preferences of users in different ages and genders with the top latent topics of each user group. Sect. 4.4.2 compares the accuracies of the proposed methods with the competitors in ad-hoc search and re-ranking on Test Collection 1. Sect. 4.4.3 presents the re-ranking performance of our methods on Test Collection 2.

In all the reported results, the symbol (*) after a numeric value denotes significant differences ($p < 0.05$, a two-tailed paired t-test) with the corresponding second best measurement. In experiments, for each user group (i.e., male users between 16-20 years old), all the 1-term, 2-term and 3-term queries are used for retrieval and evaluation. All the results presented in below are the average values over all user groups in each test collection. Notice that the search performances could be very different from groups to groups. For example, for most of the queries, the search accuracy of the 16-20_male group is higher than the 46-54_female user group.[7]

4.4.1 Qualitative Study of the Topic Model

Before presenting the search results of the retrieval methods based on the UIA-MIT model, we first examine the effectiveness of this topic model on whether

[7]The performance differences across different groups might be caused by the differences between the number of relevant songs in different groups. The number of relevant songs in groups of 46-54_male and 45-54_female is much smaller because there are fewer users and few labeled loved tracks for each user in these groups. In this study, we focus on the effects of exploiting user's age and gender information on music search performance, and thus have not presented and compared the search results between different groups.



(a) Topic preferences of different age groups (b) Topic preferences of different genders

Figure 4.5: Comparisons of the representative topics of different age ranges and groups

.

Table 4.5: The top words of the most representative topics for age and gender music preferences in the 45 topics.

Topics	Top Words
Topic 10	rock, british, **classic**, **blues**, roll, hard, pop, **60s**, radio, guitar, male, **70s**
Topic 11	rock, indie, alternative, punk, pop, male, good, post, british, dance, **00s**, garage
Topic 15	hop, hip, pop, rap, rock, dance, male, alternative, soul, classic, party, american
Topic 16	rock, alternative, indie, pop, male, brit, british, great, sad, beautiful, chill
Topic 20	rock, punk, alternative, post, indie, british, **classic**, **80s**, wave, loved, pop
Topic 35	rock, indie, pop, alternative, vocal, male, punk, folk, **00s**, acoustic, good, key
Topic 37	pop, female, dance, rock, indie, vocalists, alternative, party, electronic
Topic 38	pop, rock, female, indie, alternative, chill, electronic, hop, trip, electronica
Topic 39	rock, pop, **80s**, **classic**, male, alternative, punk, vocal, guitar, electric, loved
Topic 40	rock, guitar, **classic**, male, **blues**, roll, pop, british, folk, **70s**, oldies, **60s**

it can capture the age music preferences and gender music preferences. The results in TC1 are used for analysis. In the UIA-MIT model, age and gender music preferences are represented by the distribution of latent topics. Here we aggregate the most 3 representative topics of different groups (7 age groups and 2 gender groups) and present the preferences of each group on these topics in Fig. 4.5a and Fig. 4.5b. The top words in each topic are shown in Table 4.5. These results are obtained with the topic number $K = 45$ and training on one of the two folds (as we use two-fold cross-validation method in experiments).

Fig. 4.5a shows the general music preferences of users in different age ranges. From the figure, it is clear that users in different ages have different music preferences. Moreover, users with larger age gap have more different music preferences. For example, users in age from 41 to 54 have no common prefer-

ence on the top 3 topics with users in age from 16 to 20. Users in age groups 36-40, 41-45, and 45-54 prefer musical topics 10, 20, 39 and 40; in contrast, users with age less than 35 favors music topics 11, and 35 more. By associating with the topic semantics, we can see that topic 10, 20, 39 and 40 are related to terms *60s*, *70s 80s classic* and *blues*, which explain why relatively elder users prefer music in these topics. On contrast, topic 11 and 35 are music related to terms *00s*, which are more likely to be liked by younger users. Fig. 4.5b shows the music differences between male and female. At the aggregation level (aggregating the top topics of males and females), it can be found that both males and females primarily listen to male artists (topic 11, topic 16 and topic 35), and females listen relatively more often to females than males (e.g., topic 37). The observation is consistent with the conclusion in [12]. From the above discussion, it is safe to conclude that the UIA-MIT model can capture the influence of age and gender on user's music interests to some extent.

4.4.2 Performance on Test Collection 1 (TC1)

4.4.2.1 Retrieval Performance

Retrieval results of the proposed methods using age and/or gender information with the competitors are reported in Table 4.6. For the three types of queries, it is obviously that queries with more terms are more difficult for all methods. As can be seen, the proposed method using age and gender information (AG-MIT) generally outperforms all the other methods over all types of queries. Besides, the MIT, A-MIT and G-MIT methods obtain at least comparable performance over the GBR method, which obtains the best performance besides our proposed methods. The results demonstrate the effectiveness of the proposed retrieval methods. From the comparisons between MIT, A-MIT, G-MIT, and AG-MIT, we can see that the consideration of user's personal information can obviously improve the search performance.

Table 4.6: Retrieval performance for 1-tag, 2-tag and 3-tag queries

Method	1-Tag Query			2-Tag Query			3-Tag Query		
	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR
TAG	.164	.134	.223	.054	.041	.094	.022	.014	.055
WLC	.178	.142	.265	.058	.051	.126	.022	.023	.079
MIT	.241	.252	.404	.138	.151	.301	.116	.132	.292
GUMR	.133	.115	.221	.030	.026	.091	.015	.012	.060
G-MIT	.276	.283	.449	.139	.155	**.316**	.116	.134	**.313**
A-MIT	.250	.259	.423	.135	.151	.314	.111	.128	.296
AG-MIT	**.339***	**.335***	**.480***	**.177***	**.184***	**.316**	**.149***	**.166***	.301

Among the tested methods, TAG, WLC, and MIT are the ones without considering user information. Comparing to the TAG method, which only uses text information in retrieval, the other three methods (WLC, GBR, and MIT) using both text and acoustic features obtain better performance. The WLC method, which uses a linear combination of similarities based on TAG and acoustic features, can only slightly improve the search performance. Notice that the WLC uses the first search result of TAG as the acoustic query, the search accuracy of TAG thus affects the improvement of the WLC method. The GBR and MIT method explore both term and acoustic information by discovering and using the intrinsic correlation between the semantics of terms and the acoustic contents, and they can significantly improve the search accuracy comparing to the TAG method.

It can be seen that the G-MIT and A-MIT methods can improve the search performance over the GBR method for 1-term query. The AG-MIT method can further improve the performance for 1-term queries and obtain better performance on 2-term and 3-term queries. The results of GUMR are quite poor, because of the simple method in modeling user's music preferences in different age and gender groups. The effects of using age or gender information in retrieval are comparable, and the gender information seems slightly more effective than age information. The performance of the AG-MIT method is

obviously better than the G-MIT and A-MIT methods, indicating that the use of both age and gender information together is more effective than using them separately.

4.4.2.2 Re-ranking Performance

This section presents the re-ranking performance based on the top 100 results of different retrieval methods. The results are reported in Table 4.7, Table 4.8 and Table 4.9. In those tables, the rows starting with "-" shows the performance obtained by the corresponding baseline methods (e.g., TAG, WLC, and GBR). Overall, the results are improved greatly and significantly by the re-ranking methods, even for 2- and 3-term queries whose initial results are very poor. An interesting finding is that the initial results of the TAG method are worse than the WLC method, however, the TAG method can obtain much better re-ranking results than the WLC method by all re-ranking methods. The results indicate that the WLC method can obtain better search results in few top positions (e.g., top 10 results), while it reduces the number of relevant results in a longer list (i.e. top 100 results).

Table 4.7: Re-ranking performance of 1-tag query

Method	TAG			WLC		
	P@10	MAP	MRR	P@10	MAP	MRR
-	.164	.134	.223	.178	.142	.265
PAR	.233	.249	.422	.086	.074	.164
MIT	.458	.491	.654	.206	.233	.389
MPR	.383	.409	.572	.185	.148	.292
G-MIT	.460	.470	.619	.197	.223	.398
A-MIT	.444	.470	.632	.195	.221	.392
AG-MIT	**.479***	**.495**	**.675***	**.228***	**.242**	**.389**

The effectiveness of the re-ranking methods based on the proposed models can be observed by comparing the MIT method with the PAR method. The MIT method can improve the performance based on the results of all the three

Table 4.8: Re-ranking performance of 2-tag query

Method	TAG			WLC		
	P@10	MAP	MRR	P@10	MAP	MRR
-	.054	.041	.094	.058	.051	.126
PAR	.099	.116	.258	.036	.035	.100
MIT	.252	.271	.444	.070	.078	.185
MPR	.237	.257	.422	.063	.069	.163
G-MIT	.260	.278	.458	.069	.079	.188
A-MIT	.254	.277	.448	.077	.086	**.198**
AG-MIT	**.267**	**.294***	**.468***	**.087***	**.091***	.192

Table 4.9: Re-ranking performance of 3-tag query

Method	TAG			WLC		
	P@10	MAP	MRR	P@10	MAP	MRR
-	.022	.014	.055	.022	.023	.079
PAR	.068	.078	.191	.022	.023	.075
MIT	.188	.204	.374	.044	.047	.124
MPR	.184	.204	.373	.039	.042	.111
G-MIT	.191	.209	.385	.048	**.053**	**.134**
A-MIT	.192	.211	.383	.045	.046	.123
AG-MIT	**.200**	**.225***	**.410***	**.053**	.048	.114

methods, and the improvements are much greater than the PAR method. Notice that the MIT method explores the relevance between queries (semantic concepts or tags) and songs in a latent music interest space, which is discovered based on the music preferences of a large number of listeners. In other words, the method *leverages the collaborative knowledge of crowds to estimate the relevance between query concepts and songs - the music preference of general users on songs with respect to query concepts.* The external knowledge is complementary to the information used by the TAG, WLG, and GBR methods, which compute the relevance between the query and song only based on their contents. Consequently, using the estimated relevance between query and song based on the MIT and UIA-MIT models for re-ranking can significantly improve the search performance.

The benefits of using user information in music retrieval can be well demonstrated by the MPR method, which can improve the search performance greatly using a heuristic method - re-ranking the songs according to their popularity in user groups. For the 1-term query on $P@10$, the relative improvement by MPR achieves more than 133% and 43% over the TAG and GBR methods, respectively. The improvement over 2-term and 3-term queries is even larger. G-MIT and A-MIT methods obtain much better results than the MPR method. The AG-MIT method can further improve the performance. Comparing to the MIT method, the advantage of the G-MIT method and A-MIT method is not obviously. Notice that improvements achieved by A-MIT, G-MIT and AG-MIT are contributed by both *the learned associations between query and songs* and *the captured age and gender music preferences* in UIA-MIT. The comparable performance between the MIT, G-MIT, and A-MIT discloses that a major part of the improvement is gained from the aspect of estimating the relevance between query concepts and songs in the latent music interest space. On the other hand, the influence of age and gender are correlated to affect user's music interests. Thus, it is not optimal to use the obtained age music preference and gender music preference individually, as in the A-MIT and G-MIT methods. Therefore, the advantages of using age or gender information over the MIT method is not obviously. UIA-MIT captures the age music preference and gender music preference together and achieves consistent and better improvement over MIT. The relative improvement of the AG-MIT for the 1-term query on $P@10$ achieves more than 192% and 79% over the TAG and GBR methods, respectively.

In summary, the proposed topic model can effectively capture the associations between songs and tags, which can be effectively used in text-based music retrieval. Besides, the exploitation of user information (i.e., age and gender) in music retrieval is very useful and can significantly improve the search performance. Furthermore, the proposed UIA-MIT model can effectively capture the

general music preferences of users in different ages and genders. And the UIA-MIT based retrieval methods can effectively incorporate the user information in music retrieval to improve the search accuracy.

4.4.3 Performance on Test Collection 2 (TC2)

From the experimental results on TC1, we can observe that using our method in re-ranking can greatly improve the search results. Thus, we focus on the performance of re-ranking and only report the re-ranking performance on TC2. In TC1, the re-ranking performance based on TAG are better than or comparable to that based on WLC and PAR. As similar results are observed in TC2, we only present the performance of re-ranking performance based on TAG. Table 4.10 presents the re-ranking results of different variants on TC2. The second row shows the search results of TAG method, and 3 - 9 rows show the re-ranking results of different methods. Besides the similar observations as in TC2, we can see that user's country information (C-MIT) can also be used to improve the performance. AGC-MIT obtains the best performance, which demonstrates that the UIA-MIT model can be extended to include other user information and the utilization of more user information can obtain better performance.

4.5 Summary

In this chapter, we presented a text-based retrieval system which can leverage user's basic information (e.g., age and gender) to significantly improve the search performance. Particularly, we proposed a User-Information-Aware Music Interest Topic (UIA-MIT) model to discover the latent music interest space of general users and capture the music preferences of users in different ages and genders. In the latent space, the association between music concepts and songs can be constructed. Thus, a music retrieval method is proposed based

Table 4.10: Re-ranking performance for 1-term, 2-term and 3-term queries in TC2

Method	1-Term Query			2-Term Query			3-Term Query		
	P@10	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR
TAG	.122	.120	.165	.042	.020	.065	.022	.017	.045
MPR	.216	.277	.445	.135	.138	.085	.103	.106	.119
MIT	.220	.278	.483	.142	.175	.183	.104	.106	.117
A-MIT	.240	.318	.528	.146	.182	.189	.103	.106	.120
G-MIT	.228	.304	.500	.135	.158	.140	.104	.108	.120
C-MIT	.244	.318	.537	.144	.181	.194	.104	.107	.120
AG-MIT	.252	.330	.547	.144	.176	.184	.104	.108	.120
AGC-MIT	**.375***	**.505***	**.688***	**.160**	**.199***	**.202**	**.105**	**.110**	**.127**

on the UIA-MIT model for text-based music retrieval. The proposed method can effectively incorporate user information in retrieval. Extensive experiments have been conducted to demonstrate the advantages of exploiting user's age and gender information in music retrieval and validate the effectiveness of the proposed retrieval methods. The research results in this chapter demonstrate the importance and potential of utilizing user-specific information in music retrieval systems. As the system presented in this chapter only needs user's basic information to improve the search results, it could be used to deal with new users in the personalized music retrieval system (described in next chapter), which suffers from the cold-start problem of new users.

Chapter 5

Personalized Text-Based Music Retrieval

In the previous chapter, we present a user information aware text-based music retrieval system, which could improve the text-based search accuracy by considering the general music preferences of users in certain age or gender. In this chapter, we describe a personalized text-based music retrieval system, which takes users' personal music preferences into account. Towards the goal, a novel Dual-Layer Music Preference Topic Model (DL-MPTM) is proposed to construct latent music interest spaces and characterize the correlations and interplays between users, songs, and keywords or terms in the latent spaces. Further, based on the DL-MPTM, we develop an effective personalized music retrieval system. To verify the performance of the system, extensive experimental studies have been conducted on two large-scale public datasets to compare the proposed method with the state-of-the-art music retrieval methods and existing personalized music retrieval methods. The results show that our method significantly outperforms the state-of-the-art approaches in terms of personalized search accuracy.

5.1 Introduction

Over the past decades, empowered by fast advances in digital storage and networking, we have witnessed ever-increasing amounts of music data from various domain applications. Meanwhile, with the proliferation of mobile devices (e.g., mobile phones and laptops) and cloud-based music service, the development of personalized music information retrieval techniques has gained greatest momentum as a means to assist users to explore large-scale music collections based on "individual preference". In music retrieval, users often use several keywords to describe their music information needs and current contexts, with the expectation that the music search engine returns a list of suitable songs. Thus, text-based music retrieval (TBMR) [151, 87] has been widely used in many real applications and commercial music services. However, existing TBMR methods only consider the relevance between songs and search keywords, while generally ignoring the effects of user's personal music preference. In fact, how a user perceives a song is very subjective, depending on their emotional and culture background [148]. For example, given a query "*sad*", *whether a song is relevant* or *the relevance level of the song* with respect to "sad" is dependent on the user's personal perception on the song. Thus, for music retrieval, it is crucial to take user's personal music preference into account and effectively model the correlation among (user, song, term). The significance of leveraging user music preference has been widely recognized in the development of smart music information systems [133]. However, few researches focus on 1) investigating the effects of user music preferences on search performance improvement; and 2) designing advanced schemes to catch and model such effects and exploit them in the development of personalized music search systems.

Motivated by discussion above, we focus on developing a personalized text-based music retrieval system and studying the effects of personal music prefer-

ences on search performance. Effective integration of the user music preference to improve retrieval performance generally requires a comprehensive and reliable modeling scheme to characterize how the user music preference affects individual music information needs. To achieve the goal, we propose a novel topic model based scheme called *D*ual *L*ayer *M*usic *P*reference *T*opic *M*odel (DL-MPTM). DL-MPTM is a novel two-layer topic model, which discovers two sets of latent topics - *latent music dimensions* and *latent semantic subtopics*. In this model, a user's music preference is represented as a mixture of *latent music dimensions*, which are discovered based on the co-occurrence of songs in playlists and co-occurrence of *latent semantic subtopics* across songs. The latent semantic subtopics are the mixtures of terms. Accordingly, the correlation among (user, song, term) can be captured by the associations of the two sets of latent topics. Based on the model, a personalized text-based music retrieval method is developed.

To study the influence of personal music preferences and validate the performance of proposed system, it is important to evaluate the system performance and compare it with other competitors. A big problem is how to construct a dataset for robust system evaluation. A naive approach is to leverage the assistance of end users to manually label songs with various music concepts. However, this approach could be very expensive in terms of time and expertise. An alternative way is to leverage users' listening logs and social tags in social music services. In recent years, the rapid growth and popularity of online social music services, such as Last.fm and Pandora[1], provide excellent sources to harvest large-scale user behavior information. When interacting with the social music portals, users leave rich digital footprints containing the details about personal music listening history (e.g., which song was played by which user at what time for how long). Through analyzing the data, comprehensive information related to users' music preferences or tastes can be obtained.

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

Besides, in those social music websites, songs are tagged by users with different types of concepts, which reveal the semantic contents of the songs. The social tags in Last.fm almost cover all the concepts that users usually use to describe songs, and have been applied for text-based music search [87, 101]. The listening history of users and social tags provide us reliable sources to learn the correlations among (user, song, term), which can be used to support music search at the personal level. Accordingly, we use the social music data to examine the performance of the proposed system. In summary, the main contributions can be summarized as follows.

- Instead of conducting large-scale user study, online social music data (user listening history and music social tags) is leveraged to study the problem of *personalized text-based music retrieval*, which has been generally ignored in existing research.
- A personalized text-based retrieval method is proposed based on a novel dual-layer topic model DL-MPTM, which captures user's music preference on songs with respect to the query via the connection of two latent semantic spaces.

The remainder of this chapter is organized as follows: Section 5.2 describes the proposed topic model - DL-MPTM and Section 5.3 introduces the personalized text-based retrieval method. Section 5.4 gives the experimental configuration and Section 5.5 reports experimental results and main findings. Finally, Section 5.6 concludes this chapter with a summary.

5.2 Dual-Layer Music Preference Topic Model

The goal is to design a personalized text-based music retrieval system for searching songs, which should not only be relevant to the query but also match user's personal music preferences. Consequently, the core research problem is

how to effectively model users' music preference on songs with respect to the search keywords, namely, the correlation among (user, song, term). Users usually prefer different types of music tracks, which can be reflected from the songs he/she usually listened to. Meanwhile, people's music preferences on songs are highly associated with the semantics embodied by the audio contents of songs. Based on the semantics, user's music preferences can be extracted by analyzing the semantics of songs listened by the users. Further, given that the semantics of songs are modeled by song's contents and user-generated annotations (e.g., social tags), the correlations among (user, song, term) can be estimated. To achieve the goal, we propose a dual-layer LDA model, which *characterizes the song's semantics based on the associations between audio contents and tags* and *models user's music interests based on the songs and their semantics*. To ease understanding of the model, we firstly introduce two important concepts.

Latent Semantic Subtopics: Latent semantic subtopics (or subtopic for short) are the latent topics discovered (in the second layer - Part B in Fig. 5.1) based on the association between song's audio contents and annotations or text words. The subtopics are modeled using the multinomial distributions of audio words and text words.

Latent Music Dimensions: Latent music dimensions (or music dimensions for short) are a set of latent topics discovered (in the first layer - Part A in Fig. 5.1) based on co-occurrence of songs and their subtopic distributions. Users' music interests are modeled using the multinomial distributions of music dimensions. A music dimension is in turn a multinomial distribution of subtopics.

5.2.0.1 Model Description

Figure 5.1 illustrates the graphical representation of *D*ual *L*ayer *M*usic *P*reference *T*opic *M*odel (DL-MPTM). The model consists of two main components: Part A (the first layer) and Part B (the second layer). The second layer (Part B) is

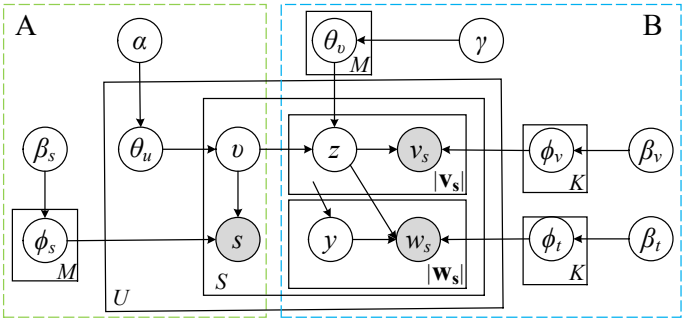


Figure 5.1: The graphical model representation of the DL-MPTM model.

a Corr-LDA model [14], which discovers subtopics based on the co-occurrence of music contents (audio words and text words) in the same song. Besides, this model discovers the associations between audio contents and text words. The first layer (Part A) is a topic model to explore music dimensions υ based on the co-occurrences of songs in the same user's profile and the subtopics associated with these songs. Each subtopic z is represented by a multinomial of audio words and a multinomial distribution of text words; each music dimension υ is represented by a multinomial distribution of songs and a multinomial distribution of subtopics. The set of subtopics in the second level is shared across different music dimensions. These subtopics, which are represented by the distribution of text words or audio words, are used to characterize the music dimensions. User music interests are represented by the multinomial distribution of music dimensions. Because the music dimension is discovered based on the co-occurrence of subtopics of songs and the subtopics are discovered based on the co-occurrence patterns of songs' contents, the dual-layer topic model discovers the latent music dimensions and subtopics in a mutual reinforcement process.

From the generative perspective, a song s with text words $\boldsymbol{w_s}$ and audio words $\boldsymbol{v_s}$ preferred by a user u, namely, an observation of $(u, s, \boldsymbol{w_s}, \boldsymbol{v_s})$, is assumed to be generated by first choosing a music dimension υ (e.g., a certain music style) from music interest $\boldsymbol{\theta_u}$ of user u. Then based on the selected topic υ, song s is drawn according to $\phi_{\upsilon,s}$, which represents the likelihood for user

u to select song s in the music dimension υ. The audio words $\boldsymbol{v_s}$ and text words $\boldsymbol{w_s}$ of song s are generated according to the subtopic distributions $\boldsymbol{\theta_\upsilon}$ of the music dimension υ. The generation process of audio words and text words is to firstly generate all the audio words, and then subsequently generate all the text words. Specifically, for each audio word v_s, a subtopic z is sampled and the audio word is generated accordingly based on ϕ_{z,v_s}. After obtaining all the audio words, for each text word, an audio word v_s is first selected and the text word w_s is generated, conditioned on the subtopic that generated the audio word. For details about the sampling process of the second layer (Part B), please refer to [14]. More formally, the process of user's profile generation is shown in Algorithm 5 (steps 4-21).

Based on the connection of two layers of topic models, DL-MPTM thus specifies the conditional joint distribution on song s and a term t given a user u and the latent variables:

$$\begin{aligned} &P(s,t|u,\boldsymbol{\theta_u},\boldsymbol{\theta_\upsilon},\boldsymbol{\phi_s},\boldsymbol{\phi_v},\boldsymbol{\phi_t}) \\ &\quad = \sum_{\upsilon=1}^{M} P(\upsilon|\boldsymbol{u},\boldsymbol{\theta_u}) P(s|\upsilon,\boldsymbol{\phi_s}) \sum_{z=1}^{K} P(z|\upsilon,\boldsymbol{\theta_\upsilon}) P(t|z,\boldsymbol{\phi_t}) \end{aligned} \tag{5.1}$$

This equation estimates how correlative user u, song s, and term t could be, and thus can be used for personalized text-based music retrieval, which is introduced in Section 5.3.

5.2.0.2 Model Inference

In the DL-MPTM model, $\alpha, \gamma, \boldsymbol{\beta_s}, \boldsymbol{\beta_v}$, and $\boldsymbol{\beta_t}$ are Dirichlet priors and predefined. The parameters needed to be estimated include: (1) user interest (user-music dimension) distribution $\boldsymbol{\theta_u}$, (2) music dimension - subtopic distribution $\boldsymbol{\theta_\upsilon}$, (3) music dimension - song distribution $\boldsymbol{\phi_s}$, (4) subtopic-term distribution $\boldsymbol{\phi_t}$ and (5) subtopic-audio word distribution $\boldsymbol{\phi_v}$. Several algorithms have been developed to approximate the parameters in variants of LDA. In our implementation, collapsed Gibbs sampling [46] is used to estimate these

Algorithm 5: Generative & Collapsed Gibbs Sampling Process for DL-MPTM

Input: A user music profile dataset D;
Number of latent topics: K; Number of latent music dimensions: M;
Dirichlet hyperparameters: $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $\boldsymbol{\alpha_g}$, $\boldsymbol{\beta_s}$, $\boldsymbol{\beta_t}$, $\boldsymbol{\beta_v}$
Output: Estimated parameters: $\boldsymbol{\theta_u}$, $\boldsymbol{\theta_\upsilon}$, $\boldsymbol{\phi_s}$, $\boldsymbol{\phi_t}$, $\boldsymbol{\phi_v}$

Initialize $\boldsymbol{Z}$ and $\boldsymbol{\Upsilon}$ by assigning random values;
Count N_u^m, N_m^s, and N_m^k based on initialized $\boldsymbol{Z}$;
Count N_k^t and N_k^v based on initialized $\boldsymbol{Z}$;
for *each latent music dimension* $\upsilon \in \{1, \cdots, M\}$ **do**
    Draw $\boldsymbol{\phi_s} \sim Dir(\cdot|\boldsymbol{\beta_s})$;
    Draw $\boldsymbol{\theta_\upsilon} \sim Dir(\cdot|\boldsymbol{\gamma})$;
for *each latent subtopic* $k \in \{1, \cdots, K\}$ **do**
    Draw $\boldsymbol{\phi_v} \sim Dir(\cdot|\boldsymbol{\beta_v})$;
    Draw $\boldsymbol{\phi_t} \sim Dir(\cdot|\boldsymbol{\beta_t})$;
for *each user* $u \in \mathcal{U}$ **do**
    Draw $\boldsymbol{\theta_u} \sim Dir(\cdot|\boldsymbol{\alpha})$;
for *each Gibbs sampling iteration* **do**
    for *each user* $u \in \mathcal{U}$ **do**
        for *each song* $s \in D_u$ **do**
            Draw a latent music dimension υ from the music interest distribution of user $\boldsymbol{\theta_u}$;
            for *each audio word* $v_s \in \boldsymbol{v_s}$ **do**
                Draw z_{v_s} from the distribution $\boldsymbol{\theta_\upsilon}$ of latent subtopics in the latent music dimension υ;
                Draw v_s from the audio word distribution $\boldsymbol{\phi_v}$ from subtopic z ;
            for *each word* $w \in \boldsymbol{w_s}$ *in the song (suppose there are* n *audio words in this song, and let* z_i *denote the sampled topic for the* i*-th audio word in previous step)* **do**
                Draw $y \sim \text{Unif}(1, 2, ..., n)^2$;
                Draw w_s from the text word distribution $\boldsymbol{\phi_t}$ from the latent subtopic z_y;
            Update N_k^t and N_k^v according to $z_{v_s} = k$, $\boldsymbol{w_s}$, and $\boldsymbol{y}$;
        Update N_u^m, N_m^s, and N_m^k according to $\upsilon_s = m$ and $z_{v_s} = k$;
Estimate model parameters $\{\boldsymbol{\theta_u}, \boldsymbol{\theta_\upsilon}, \{\boldsymbol{\phi_s}, \boldsymbol{\phi_t}$, and $\boldsymbol{\phi_v}\}$ according to Eq. 5.6, Eq. 5.7 and Eq. 5.8, respectively.

parameters, as this method has been successfully applied in many large scale applications of topic models [46, 103]. Notice that in the learning of a model, Gibbs sampling iteratively updates each latent variable given the remaining variable until it converges. The Collapsed Gibbs Sampling process of DL-MPTM is described in Algorithm 5.

Given a user music profile corpus D with user set U, for each user $u \in U$, a playlist $\{s_1, s_2, ..., s_n\}$ records his/her playing behaviors or music profile. Each song s contains a sequence of text words $\boldsymbol{w_s}$ and a sequence of audio word $\boldsymbol{v_s}$. In the Gibbs sampling process, the playlists of users are sampled in sequence. Let $\boldsymbol{S}$ be the sampling sequence in the Gibbs sampling process, which is the concatenation of songs in the playlists of all the users. Similarly, let $\boldsymbol{V}$ and $\boldsymbol{W}$ denote the corresponding sampling sequences of audio words and text words. $\boldsymbol{\Upsilon}$ and $\boldsymbol{Z}$ denote the set of latent music dimensions and subtopics corresponding to the song sequence and audio words sequence, respectively. Besides, $\boldsymbol{Y}$ is the assignment indicators of the word sequence $\boldsymbol{W}$. $\boldsymbol{S_{\neg i}}$ denotes $\boldsymbol{S}$ excluding the i-th song s_i in $\boldsymbol{S}$. Similar notation is used for other variables. For the sampling of latent music dimension $\upsilon_i = l$ for s_i, the probability is

$$P(\upsilon_i = l|\boldsymbol{\Upsilon_{\neg i}}, \boldsymbol{S}, \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{V}, \boldsymbol{W}) \propto \frac{\alpha_l + N^l_{u,\neg i}}{\sum_{l=1}^{L}(N^l_{u,\neg i} + \alpha_l)} \cdot \frac{\beta_s + N^s_{l,\neg i}}{\sum_{s=1}^{M}(N^s_{l,\neg i} + \beta_s)} \cdot PLS(l, s_i) \tag{5.2}$$

$$\begin{aligned} PLS(m, s_i) &= \frac{\prod_{k=1}^{K}\Gamma(\gamma_k + N^k_l)}{\Gamma(\sum_{k=1}^{K}(N^k_l + \gamma_k))} \cdot \frac{\Gamma(\sum_{k=1}^{K}(N^k_{l,\neg i} + \gamma_k))}{\prod_{k=1}^{K}\Gamma(\gamma_k + N^k_{l,\neg i})} \\ &= \frac{\prod_{k=1}^{K}(\gamma_k + N^k_l - 1)!}{\prod_{k=1}^{K}(\gamma_k + N^k_l - n_{l,k,s_i} - 1)!} \cdot \frac{(\sum_{k=1}^{K}(N^k_{l,\neg i} + \gamma_k - n_{l,k,s_i}) - 1)!}{(\sum_{k=1}^{K}(N^k_{l,\neg i} + \gamma_k) - 1)!} \end{aligned} \tag{5.3}$$

where N^l_u denotes the number of times that music dimension l is observed in u's playlist. N^k_l is the number of times that subtopic k is observed in music dimension l. Notice that the exclusion of $\upsilon = l$ will cause the changes of

N_l^k for all $k = [1, K]$. $N_{l,\neg i}^k$ denotes the number of times latent subtopic k is observed in latent music dimension l by excluding l assigned to song s_i, and $N_{l,\neg i}^k = N_l^k - n_{l,k,s_i}$. n_{l,k,s_i} denote the number of times the subtopic k is observed in music dimension l due to s_i. $PLS(l, s_i)$ denotes the effects of the exclusion of $\upsilon = l$ on the distribution of subtopics in the music dimension l. $\Gamma(\cdot)$ is the Gamma function.

Next we introduce the sampling of subtopic $z_j = k$ for an audio word $v_j = v$ in s_i and the sampling of all text words of the song s_i. Notice that the text words in s_i are sampled after sampling all the audio words in s_i, as the assignment of z_j to the words in a song is dependent on the subtopic sequence of audio words in this song. The probability of $z_j = k$ to an audio word $v_j = v$ is:

$$P(z_j = k|\boldsymbol{\Upsilon}, \boldsymbol{S}, \boldsymbol{Z}_{\neg \boldsymbol{j}}, \boldsymbol{Y}, \boldsymbol{V}, \boldsymbol{W}) \propto \frac{\gamma_k + N_{l,\neg j}^k}{\sum_{k=1}^K (N_{l,\neg j}^k + \gamma_k)} \cdot \frac{\beta_v + N_{k,\neg j}^v}{\sum_{v=1}^V (N_{k,\neg j}^v + \beta_v)} \cdot PZ(k) \tag{5.4}$$

$$\begin{aligned} PZ(k) & \frac{\prod_{t=1}^T \Gamma(\beta_t + N_k^t)}{\Gamma(\sum_{t=1}^T (N_{k,\neg j}^t + \beta_t))} \cdot \frac{\Gamma(\sum_{t=1}^T (N_k^t + \beta_t))}{\prod_{t=1}^T \Gamma(\beta_t + N_{k,\neg j}^t)} \\ & = \frac{\prod_{t=1}^T (\beta_t + N_k^t - 1)!}{\prod_{t=1}^T (\beta_t + N_k^t - n_t - 1)!} \cdot \frac{(\sum_{t=1}^T (N_{k,\neg j}^t + \beta_t - n_t) - 1)!}{(\sum_{t=1}^T (N_{k,\neg j}^t + \beta_t) - 1)!} \end{aligned} \tag{5.5}$$

where N_k^v is the number of times that subtopic $z_j = k$ is assigned to audio word $v_j = v$. $N_{k,\neg j}^t$ denotes the number of times that t is assigned to subtopic k before assigning k to the j-th audio word of song s_i, and $N_{k,\neg j}^t = N_k^t - n_t$. n_t denotes the number of times that term t is assigned to the subtopic of the j-th audio word in the current song s_i. Notice that the exclusion of $z_j = k$ for audio word v_j may influence the assignment of $z_j = k$ to multiple text terms and multiple times. Similar to $PLS(l, s_i)$, $PZ(k)$ denotes the effects of the exclusion of $z_j = k$ on the distribution of text terms in the subtopic k.

Based on the state of the Markov chain $\boldsymbol{\upsilon}$ and $\boldsymbol{z}$, we can estimate the

parameters:

$$\theta_{u,m} = \frac{\alpha_m + N_u^m}{\sum_{m=1}^{M}(\alpha_m + N_u^m)} \qquad \theta_{m,k} = \frac{\gamma_k + N_m^k}{\sum_{k=1}^{K}(\gamma_k + N_m^k)} \tag{5.6}$$

$$\phi_{m,s} = \frac{\beta_s + N_m^s}{\sum_{s=1}^{M}(\beta_s + N_m^s)} \qquad \phi_{k,t} = \frac{\beta_t + N_k^t}{\sum_{t=1}^{T}(\beta_t + N_k^t)} \tag{5.7}$$

$$\phi_{k,v} = \frac{\beta_v + N_k^v}{\sum_{v=1}^{V}(\beta_v + N_k^v)} \tag{5.8}$$

5.3 Retrieval Model

The goal of the retrieval model is to search a subset of songs that are relevant to a particular query. Let $q = \{t_1, t_2, ..., t_n\}$ represent user u's query consisting of n terms. The retrieval algorithm aims at ranking songs based on their relevance to the query according to u's music preference on the songs. Notice that the relevance level of a song with respect to a query is dependent on user's music taste. Given a query q issued by user u, for a song s, $P(s|q,u)$ denotes the likelihood or probability of user u preferring this song s with respect to the query q. Thus, candidate songs can be ranked in the descending order of their probabilities $P(s|q,u)$ with respect to the user and query (u,q). According to Bayes rule, $P(s|q,u)$ can be computed as:

$$P(s|q,u) = \frac{P(q,s|u)P(u)}{P(q,u)} \propto P(q,s|u) \tag{5.9}$$

where $P(q,s|u)$ represents the relevance of song s to query q based on user u's opinions on the song.

With the posterior estimation of $\boldsymbol{\theta_u}$, $\boldsymbol{\theta_\upsilon}$, $\boldsymbol{\phi_s}$, and $\boldsymbol{\phi_t}$ in the DL-MPTM, we have:

$$\begin{aligned} P(q,s|u,\boldsymbol{\theta_u},\boldsymbol{\theta_\upsilon},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) &= \sum_{\upsilon=1}^{M} P(\upsilon|u,\boldsymbol{\theta_u}) P(q,s|\upsilon,\boldsymbol{\theta_\upsilon},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) \\ &= \sum_{\upsilon=1}^{M} P(\upsilon|u,\boldsymbol{\theta_u}) \prod_{i=1}^{n} P(t_i,s|\upsilon,\boldsymbol{\theta_\upsilon},\boldsymbol{\phi_s},\boldsymbol{\phi_t}) \end{aligned} \tag{5.10}$$

where $P(\upsilon|u, \boldsymbol{\theta_u})$ is the probability of user u selecting music dimension υ, and $P(q, s|\upsilon, \boldsymbol{\theta_\upsilon}, \boldsymbol{\phi_s}, \boldsymbol{\phi_t})$ is the joint probability of query q and s in the music dimension υ. In the derivation, we assume the query terms are independent from each other under this specific music dimension. Given the music dimension υ, s and t are independent, the joint probability of term t_i and song s in the music dimension υ can be estimated by multiplying the the probability of s and t_i in the music dimension υ: $P(s|\upsilon, \boldsymbol{\phi_s})$ and $P(t_i|\upsilon, \boldsymbol{\theta_\upsilon}, \phi_t)$.

$$P(t_i, s|\upsilon, \boldsymbol{\theta_\upsilon}, \boldsymbol{\phi_s}, \boldsymbol{\phi_t}) = P(s|\upsilon, \boldsymbol{\phi_s}) \sum_{z=1}^{K} P(t_i|z, \boldsymbol{\phi_t}) P(z|\upsilon, \boldsymbol{\theta_\upsilon}) \tag{5.11}$$

The probability of term t_i in music dimension υ can be obtained by the generative probability of term t_i in the subtopic space: $\sum_{z=1}^{K} P(t_i|z, \boldsymbol{\phi_t}) P(z|\upsilon, \boldsymbol{\theta_\upsilon})$. Based on Eq. 5.10 and Eq. 5.11, the probability of user u selecting s for query q can be estimated:

$$\begin{aligned} P(q, s|u, \boldsymbol{\theta_u}, \boldsymbol{\theta_\upsilon}, \boldsymbol{\phi_s}, \boldsymbol{\phi_t}) &= \sum_{\upsilon=1}^{M} P(\upsilon|u, \boldsymbol{\theta_u}) \prod_{i=1}^{n} P(s|\upsilon, \boldsymbol{\phi_s}) \sum_{z=1}^{K} P(t_i|z, \boldsymbol{\phi_t}) P(z|\upsilon, \boldsymbol{\theta_\upsilon}) \\ &= \sum_{\upsilon=1}^{M} \theta_{u,\upsilon} \cdot \prod_{i=1}^{n} \phi_{\upsilon,s} \cdot \sum_{z=1}^{K} \theta_{\upsilon,z} \cdot \phi_{z,t_i} \end{aligned} \tag{5.12}$$

Intuitively, for a specific music dimension υ, $P(\upsilon|u, \boldsymbol{\theta_u})$ denotes the preference of user u in this dimension; $P(s|\upsilon, \boldsymbol{\phi_s})$ denotes the likelihood of song s in this dimension; $\sum_{z=1}^{K} P(z|\upsilon, \boldsymbol{\theta_\upsilon}) P(t|z, \boldsymbol{\phi_t})$ denotes the likelihood of a term t in this dimension. Thus, $P(\upsilon|u, \boldsymbol{\theta_u}) P(s|\upsilon, \boldsymbol{\phi_s}) \sum_{z=1}^{K} P(z|\upsilon, \boldsymbol{\theta_\upsilon}) P(t|z, \boldsymbol{\phi_t})$ indicates the likelihood for user u to consider song s is relevant to term t in this music dimension.

Algorithm 6 summarizes the whole procedure of personalized text-based retrieval method. DL-MPTM training process can be carried out in offline phase (line 1 - 2). Personalized music search is based on the obtained parameters in DL-MPTM, for a given query q, a rank list $\mathcal{L}$ can be returned (line 3 - 4).

Algorithm 6: DL-MPTM based personalized text-based music retrieval

Offline Phase: DL-MPTM model training
Input: Corpus D with music profiles of users $\mathcal{U}$,
Output: $\theta_{u,\upsilon}$, $\phi_{\upsilon,s}$, $\theta_{\upsilon,z}$ and ϕ_{z,t_i}

1: Train the DL-MPTM model using the collapsed Gibbs sampling method described in Algorithm 5.
2: Estimate $\theta_{u,\upsilon}$, $\phi_{\upsilon,s}$, $\theta_{\upsilon,z}$ and ϕ_{z,t_i} using Eq. (6) - Eq.(8)

Online Phase: personalized music search
Input: A query $q = \{t_1, t_2, ..., t_n\}$
Output: A ranking list $\mathcal{L}$

3: Compute $P(q, s|u)$ using Eq. 5.12 based on the estimate parameters
4: Sort the songs into a ranking list $\mathcal{L}$ in the descending order of their probabilities $P(q, s|u)$

Discussion. When the system is used in real applications, it will face two problems at the early stage: (1) how to train the model when only a small number of users is available at the early stage; and (2) how to deal with the cold-start problem of new users. For the first problem, the abundant online users music listening behaviours could be harvested and leveraged to train the model. For the second problem, a common method is to ask users to provide some initial data at the beginning, such as their favourite songs. However, the input of a long list of songs will bring a heavy burden to users. Alternatively, in our system, we could leverage users basic information, such as age, gender, and country information to provide personal service at the early stage, namely, using the system described in Chapter 4. As demonstrated in Chapter 4, the consideration of age, gender and country could greatly improve the search performance, which will give users to have a good impression on the system at the beginning.

5.4 Experimental Configuration

In this section, we present the experimental settings for the performance evaluation, including test collections, query set with corresponding ground truth, competitors and performance metrics.

5.4.1 Test Collections

In order to achieve good repeatability of the experiments, test collections are developed based on two public datasets. Their details are as follows.

Taste Profile Subset (TPS) [3] [**97**]: This dataset consists of more than 48 million triplets (*user, song, count*) gathered from user listening histories. Here, "(user, song, count)" refers to the number of times (*i.e., count*) the *user* played the *song*. It contains approximately 1.2 million unique users and covers more than 380,000 songs. From this dataset, we randomly select 10,000 users with their listening records for our experiments.

Lastfm-Dataset-1K (Lastfm-1K)[4] [**50**]: This dataset contains (*user, timestamp, artist, song*) quadruples collected from the Last.fm using the public API. This dataset includes the listening history (until May 5th, 2009) of 992 users, 961,417 songs of 176,948 artists. Based on the quadruples records, we can also get the triplets (*user, song, count*) for this dataset.

In order to ensure the quality of test collections, the p-core filtering method [9] is used to filter users and songs. The p-core of level k has the property, that each song was listened to by at least k users and each user listened to at least k songs. In the experiments, k is set to 20. For the remaining songs, the 30 seconds audio samples were downloaded from 7digital[5], and their tags were crawled from Last.fm. Table 5.1 summarizes the details about the two datasets used in experiments. It is worth mentioning that two datasets have very different properties. Comparing with TPS, Lastfm-1K contains less users while each user has richer listening records. Thus, two datasets are used to examine the performances of personalized music retrieval systems in two scenarios: (1) with rich users' listening records available (Lastfm-1K), and (2) with limited users' listening records available (TPS), respectively.

[3] http://labrosa.ee.columbia.edu/millionsong/tasteprofile

[4] http://www.dtic.upf.edu/ ocelma/MusicRecommendation Dataset/lastfm-1K.html

[5] https://www.7digital.com/

Table 5.1: Details of two datasets used in experiments.

Dataset	#User	#Songs	#Artist	#Ave. Listened Songs per User
Lastfm-1K	992	7433	881	335.51
TPS	7022	2332	1094	15.96

The training of DL-MPTM model needs the played records of songs by users and the songs' contents, including textual content (e.g., textual words describing the song) and music content (e.g., audio words of the song). To facilitate the DL-MPTM training, we organize the related data into three types of documents. The description and generation process of the three types of documents are presented below.

User-Song Document For each user, a user-song document is generated based on his/her played records. The document is comprised by the concatenation of the songs (a "song" in a document is indexed by a unique ID) played by the users. For example, if a user u with profiles $(u, s_1, 2)$, $(u, s_2, 3)$, $(u, s_3, 1)$, the user's user-song document is $\{s_1,\ s_1,\ s_2,\ s_2,\ s_2,\ s_3\}$. It is worth noticing that the songs in the documents can be in any order of sequence. To accelerate the training process, the user-song document for each user is created by concatenating the songs that were played more than 2 times by the user, and each song only appears once. Thus, for each user, the user-song document is actually a playlist consisting of the songs that were preferred by the user in the past. For the users who are used as query users in experiments, half of the songs in their playlists are randomly selected as test songs and thus removed from the user-song document used in the training stage (see Sect. 5.4.2).

Song-Text Document The document contains the textual contents of the song, namely, the text words of a song used in the DL-MPTM. In our implementation, social tags are used to represent the text documents of songs. Our model is to capture the correlation of user, song, and term to facilitate personalized search. The tags of each song are collected from Last.fm using

public API (Track.getTopTags). In our implementation, for each dataset, we filtered the tags that appeared in less than 10 songs. Besides, we also remove the tags which express personal preferences on the songs, such as "favorite songs", "favorite", "best song forever", etc. The remaining tags of a song are concatenated together and tokenized with a standard stop-list to form the text document for the song.

Song-Audio Document The document contains the audio content of a song, namely, the audio words used in the DL-MPTM. The audio contents of one song are represented by "bag-of-audio-words" document. An audio word is a representative short frame of audio stream in a music corpus. The general procedures to generate the audio words consists of three steps: (1) segment the audio track of each song in a corpus into short frames; (2) extract acoustic features from each short frame; and (3) apply a clustering algorithm (e.g., k-means) to group the short frames into n clusters based on their acoustic features. The cluster centers are the audio words generated for the corpus. By encoding each short frame of a song with the nearest cluster center (or audio word), then the song is indexed as a sequence of audio words. In our implementation, we segment each song into 0.05s short frames without overlapping. Also, each song is converted to a standard mono-channel and 22,050 Hz sampling rate WAV format. Mel Frequency Cepstral Coefficients (MFCCs) [91] feature is used to generate the audio words. For each frame, a 13-d MFCCs vector with its first and second instantaneous derivatives are extracted, achieving a final 39-d MFFCs feature. We use K-means to generate the audio words. And for each dataset, we generate a vocabulary of 4096 audio words.

5.4.2 User-Specific Query, Test Collection and Ground Truth

In personalized music retrieval, a positive result should not only be relevant to the query but also be preferred by the query user[6]. In other words, to evaluate personalized music retrieval systems, we need to know (1) whether the result is relevant to the query, and (2) whether the user prefers the result. Therefore, user's preferences on all the songs in the test collection should be available in the evaluation. To achieve the goal, we create the query set and the test collection specific to each individual user. Firstly, a set of users is randomly selected from the datasets (Lastfm-1K and TPS) as query users. Then, for each user, a set of text queries are generated and a test collection for this specific user is created by randomly sampling half of the songs from his/her user-song document. In the user-specific test collection, the played times of songs can be used to estimate the user's preferences on these songs. Specifically, the relevance levels of a song with respect to a user-specific query are defined as follows.

- Non-relevant (0): song's text document does not contain all the query terms *or* the user listened to the song only once.
- Relevant (1): song's text document contains all the query term, *and* the user listened to the songs for 2 to 5 times.
- Highly relevant (2): song's text document contains all the query term, *and* the user listened to the songs for more than 5 times.

The definitions of relevance levels are based on the assumption that more times a user listens to a song, the higher preference level the user have on the song. The evidence that a user listened to a song more than two times

[6]The user who submits the query is called the query user. In personalized information retrieval, user and query should be in pairs. Afterward, we use "query users" to refer to the users used in the search stage.

Table 5.2: Several examples for three types of queries.

1-Word Query	2-Word Query	3-Word Query
rock	chill, soft	00s, male, rock
metal	chill, mellow	00s, indie, mellow
piano	male, mellow	chill, mellow, rock
happy	drums, guitar	british, male, rock
rainy	country, guitar	melancholy, rock, sad
driving	danceable, harmonies	mellow, folk, tonality
energetic	alternative, guitar	guitar, rock, vocalists
romantic	emotional, romantic	chillout, mellow, rock

indicates that the user shows some interests in the song. The songs listened to only once are regarded as irrelevant, since it could be a variety of reasons why users listen to a song only once. Notice that for a user, his/her listened songs, which are used in the *user-song document* in the topic model training stage, are removed from the test collections in the retrieval stage.

To test the performance of queries used in real scenarios, three types of text queries are developed for evaluation purpose: one-, two- and three-word queries, as users seldom issue long queries for music search in reality [100]. This strategy is also often applied in previous text-based music retrieval studies [100, 151]. For the one-word queries in each dataset, the most frequently used words are used as candidates. For the two- and three-word queries, the most frequent co-occurrent two and three words in tags are used as candidates, respectively. The *query users* and *user-specific queries* are carefully selected from these candidates to ensure that, for each user, the user-specific test collection contains sufficient relevant songs for his/her queries (for the fair comparisons of different retrieval methods) [100]. The query words cover the commonly used music concepts, such as *genre*, *instrument*, *mood*, and *era*. Table 5.2 shows the query examples used in the experiments.

Since the average number of songs played by users in two datasets are very different, different numbers of users and queries can be generated in two

datasets. The details about users and queries in both datasets are as below.

- **Lastfm-1K**: In this dataset, 124 users are selected as query users, and 96 different queries (30 one-word queries, 30 two-word queries, and 36 three-word queries) are selected. The selected queries are the same for all the users. The number of songs in this test collection of each user is at least 500. In total, there are 11,904 user-specific queries used in this dataset.

- **TPS**: Because the number of songs listened by users in this dataset is much smaller, few user-specific queries can be applied in order to make sure that there are enough positive songs (for each query) in the user-specific test collections. Finally, we select 20 users and 20 queries (8 one-word queries, 6 two-word queries, and 6 three-word queries) per user. Similarly, the queries of all the users are the same. The least number of songs in the test collection for each user is set to be 100. In total, there are 400 user-specific queries used in this dataset.

5.4.3 Experimental Setup

The section introduces the details about competitors, evaluation metrics and system parameters.

Competitors To verify the effectiveness of the proposed personalized text-based music retrieval system, we compare it with three text-based music retrieval (TBMR) methods and an existing personalized music retrieval (PMR) method. The three TBMR methods are tag-based music retrieval (TAG), weighted linear combination (WLC), and post-hoc audio-based reranking (PAR), which are described in Sect. 4.3.2.1. The PRM method is proposed in [48]. The topic model captures the text associations based on their co-occurrences in the same song and the songs associations based on their co-occurrences in the same user's profile under the same latent space. Each user music preference is

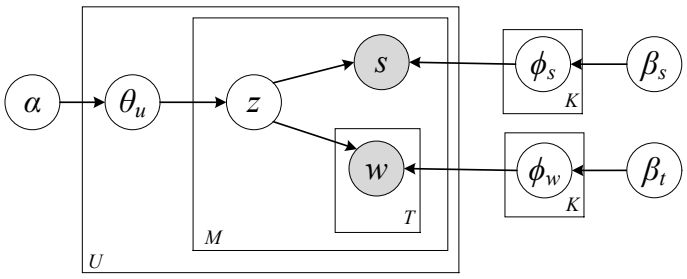


Figure 5.2: Graphical representation for PRM.

modeled as a multinomial distribution over a set of topics and each topic has a distribution over a set of songs and texts. This model does not take the music contents into account.

Evaluation Metrics In information retrieval, users are more interested in results in the top positions. Therefore, we focus on the evaluation of top results in terms of accuracy. Several standard information retrieval metrics are used, including precision at k (Precision@k), Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain at k (NDCG@k) [57]. The relevance levels (i.e., 0, 1, and 2) are used to compute NDCG. For Precision@k and MAP, both relevant (i.e., 1) and highly-relevant (i.e., 2) results are regarded as positive results.

Parameter Setting In our implementation, the Dirichlet hyper-parameters of both topic models (DL-MPTM and PRM) are empirically set: $\alpha = 1.0$, $\gamma = 1.0$, $\beta_s = \beta_t = \beta_v = 0.01$. We carefully tune the latent topic numbers in both topic models. In DL-MPTM, the number of latent music dimension is tuned in $\{5, 10, 20, 30, 40, 50, 60\}$ and the number of latent sub-topics is tuned in {20, 40, 60, 80, 100, 150}. The number of latent topics in PRM is tuned in {20, 40, 60, 80, 100, 150}. Besides, the combination weight w in WLC retrieval methods are both tuned from 0 to 1 in steps of 0.1.

5.5 Experimental Results

This section reports the experimental results of our methods and other competitors on retrieval performance. The reported results of DL-MPTM and PRM are based on the optimal numbers of latent topics in each dataset. The reported results based on MAP and NDCG in all the tables are truncated at 10, namely, MAP@10 and NDCG@10. The symbol (*) after a numeric value denotes significant differences ($p < 0.05$, a two-tailed paired t-test) with the corresponding second best measurement. All the results presented in below are the average values of queries over all the users.

5.5.1 Retrieval Performance

5.5.1.1 Effectiveness

Table 5.3, Table 5.4 and Table 5.5 report the retrieval performance on different queries consisting of one, two and three words on the two datasets, respectively. As can be seen, the proposed model outperforms all the other algorithms over both datasets. Larger performance gain can be achieved when considering more results in the top positions; in particular, the improvements in P@10, MAP and NDCG are statistically significantly compared to other algorithms. After comparing the results gained using TAG, WLC, and PAR, it is easy to find that for Lastfm-1K dataset, the consideration of audio features with text can improve the search results. Besides, PAR obtains better results than WLC does. However, in TPS dataset, the performance decreases when using acoustic features for re-ranking. It is worth noticing that the results of WLC in TPS are the same to TAG. It is mainly because the best performance of WLC can be achieved by only using text feature in WLC. Notice that the size of search dataset in Lastfm-1K is much larger than that of users in TPS (see Sect. 5.4.2). Generally, it is difficult for the content-based method to achieve better search results over a smaller dataset, because finding songs with similar

contents in smaller datasets is harder. Thus, the audio feature does not work well when being applied to support search over the TPS dataset.

Table 5.3: Retrieval performance for 1-word queries

Dataset	Metric	TAG	WLC	PAR	PRM	DL-MPTM
Lastfm-1K	P@3	.669	.662	.741	.840	**.842**
	P@5	.671	.654	.735	.818	**.851***
	P@10	.673	.658	.715	.792	**.858***
	MAP	.668	.669	.728	.824	**.852**
	NDCG	.481	.495	.548	.611	**.663***
TPS	P@3	.550	.550	.492	.648	**.667***
	P@5	.600	.600	.470	.609	**.640***
	P@10	.557	.557	.455	.583	**.633***
	MAP	.558	.558	.477	.621	**.645**
	NDCG	.434	.434	.365	.492	**.520***

Table 5.4: Retrieval performance for 2-word queries

Dataset	Metric	TAG	WLC	PAR	PRM	DL-MPTM
Lastfm-1K	P@3	.664	.652	.734	.839	**.839**
	P@5	.657	.646	.724	.823	**.845**
	P@10	.664	.653	.713	.795	**.853**
	MAP	.660	.658	.720	.826	**.846**
	NDCG	.483	.489	.541	.602	**.663***
TPS	P@3	.533	.533	.483	.633	**.658***
	P@5	.575	.575	.495	.565	**.635***
	P@10	.565	.563	.450	.545	**.629***
	MAP	.564	.565	.468	.595	**.639***
	NDCG	.425	.424	.362	.479	**.534***

On the other hand, PRM and DL-MPTM achieve much better performance than the other three methods (TAG, PAR, and WLC), which have not taken the personal music preferences into account. It demonstrates the importance of user's music preference in facilitating effective music retrieval. DL-MPTM's performance improvement over PRM on both datasets demonstrates the effectiveness of our proposed dual-layers topic model in capturing the correlation among (user, song, term). In PRM, the correlation is modeled using the same

Table 5.5: Retrieval performance for 3-word queries

Dataset	Metric	TAG	WLC	PAR	PRM	DL-MPTM
Lastfm-1K	P@3	.643	.675	.714	.827	**.846**
	P@5	.647	0669	.710	.814	**.848***
	P@10	.656	.669	.696	.789	**.848***
	MAP	.649	.650	.704	.819	**.851***
	NDCG	.482	.486	.551	.611	**.664***
TPS	P@3	.490	.490	.486	.567	**.655***
	P@5	.526	.526	.451	.555	**.625***
	P@10	.533	.531	.443	.557	**.613***
	MAP	.516	.516	.469	.574	**.619***
	NDCG	.412	.411	.363	.486	**.517***

latent space, which is discovered based on both the co-occurrence of songs in playlists and the co-occurrence contents of songs. In DL-MPTM, the correlation is captured by two layers of connected latent spaces: the low-level latent space (constructed by *latent semantic subtopics*) is discovered based on the co-occurrence contents of songs, the high-level latent space (constructed *latent music dimensions*) is discovered based on the co-occurrence of songs in playlists and the co-occurrence of latent subtopics across songs.

Table 5.6 compares the performances of TAG, PAR, and DL-MPTM based on the top five search results in the ranking lists of one representative query in each type. The relevance level of each song in the top five positions is also shown. The results demonstrate that DL-MPTM achieves much better performance in task of searching user preferred songs with respect to the queries, comparing to TAG and PAR methods. For example, in response to the query "*guitar, pop*", DL-MPTM places three high-relevant songs at the top rank, compared with only one ranked by the TAG model at the 5th position and two ranked at the 3rd and 4th positions by the PAR model.

Table 5.6: The top 5 songs in the ranking lists obtained by the TAG, PAR, and DL-MPTM models for 3 representative queries of a user "*user_000477*". The relevance level of each result is shown in the parentheses after each result, e.g., "(2)" indicates high relevance (see Sect. 5.4.2).

Query	Modal	Top 5 Songs
Metal	TAG	System of a Down - thetawaves (1)
		System of a Down - I-E-A-I-A-I-O (1)
		Linkin Park - Valentine's day (2)
		Korn - Did my time (1)
		Metallica - Nothing Else Matters (1)
	PAR	Rage Against the Machine - Bullet in the head (1)
		Linkin Park - Valentine's day (2)
		Audioslave - Set it off (1)
		Goldfrapp - Cologne cerrone houdini (1)
		Incubus - Anna molly (2)
	DL-MPTM	**Rammstein - Du hast** (2)
		A Perfect Circle - Over (1)
		Nirvana - Smells like teen spirit (2)
		Muse - Hysteria (1)
		AC/DC - Back In Black (2)
Guitar,pop	TAG	Dread Zeppelin - Misty mountain hop (0)
		Dire Straits - Sultans of swing (1)
		Dire Straits - Money for nothing (1)
		Dread Zeppelin - Your time is gonna come (0)
		Dire Straits - Brothers in arms (2)
	PAR	New Order - crystal (0)
		Dire Straits - Romeo and juliet (1)
		Linkin Park - Valentine's day (2)
		The Smashing Pumpkins - 1979 (2)
		Red Hot Chili Peppers - Mellowship slinky in b major (1)
	DL-MPTM	**Oasis - Wonderwall** (2)
		The Smashing Pumpkins - 1979 (2)
		The Cranberries - Zombie (2)
		Blur - Song 2 (1)
		Oasis - Live forever (1)
Guitar, rock, vocalists	TAG	Lez Zeppelin - Communication breakdown (0)
		Dread Zeppelin - Misty mountain hop (0)
		Lez Zeppelin - Whole lotta love (0)
		Dire Straits - Sultans Of Swing (1)
		Dire Straits - Money for nothing (1)
	PAR	The Smiths - Stretch out and wait (0)
		New Order - Crystal (0)
		Interpol - The heinrich maneuver (0)
		Klaxons - Two receivers (0)
		Dire Straits - Romeo and juliet (1)
	DL-MPTM	**AC/DC - Back in black** (2)
		AC/DC - Highway to hell (2)
		Dread Zeppelin - Heartbreaker (0)
		The Cranberries - Zombie (2)
		AC/DC - Hells bells (2)

Table 5.7: Retrieval results for query categories. The best results for each category are indicated in bold.

Category	Metric	TAG	WLC	PAR	PRM	DL-MPTM
Emotion	P@10	.684	.659	.721	.790	**.863**
	MAP	.677	.672	.732	.822	**.858**
	NDCG	.487	.498	.550	.606	**.668**
Genre	P@10	.649	.646	.701	.793	**.852**
	MAP	.639	.644	.718	.831	**.836**
	NDCG	.463	.474	.542	.616	**.647**
Instrument	P@10	.673	.651	.724	.803	**.871**
	MAP	.665	.669	.735	.835	**.862**
	NDCG	.474	.491	.552	.620	**.673**
Vocals	P@10	.657	.665	.717	.796	**.842**
	MAP	.651	.672	.720	.830	**.841**
	NDCG	.473	.488	.551	.623	**.648**
Others	P@10	.685	.671	.711	.787	**.850**
	MAP	.687	.683	.729	.808	**.847**
	NDCG	.497	.518	.547	.603	**.666**

5.5.1.2 Robustness

By comparing the results of different query types (one-, two- and three-word queries), we can observe that the search performance is slightly decreased when the query complexity increases. For different types of queries, DL-MPTM achieves significant and consistent improvement over all metrics, showing a superior robustness across multi-word queries.

Music is usually described by different categories of music concepts, such as *mood*, *instrument*, *genre*, and *vocals*, which have been widely studied in music retrieval related research, such as classification and annotation. We examine the search performance of our method over other methods on different categories of music concepts. Table 5.7 presents evaluation results. One-word queries are classified into different music concept categories as shown in the table. We focus on the one-word queries, since the two- and three-word queries could be the combination of different categories. The category "Other" con-

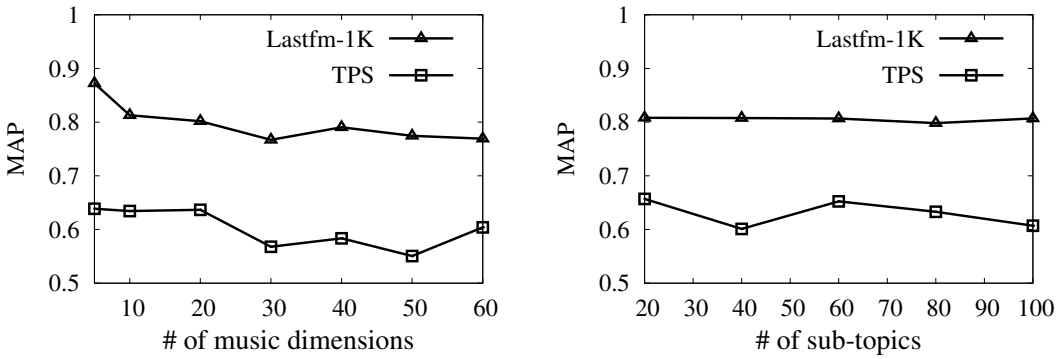


Figure 5.3: Effects of the number of latent topics in topic model based retrieval methods.

tains queries, such as "*driving*", "*slow*", "*sexy*", which cannot be classified into other four categories. The significant improvements over other methods on P@10, MAP and NDCG show the effectiveness and robustness of DL-MPTM over different music concept categories.

Comparing the search performances of all the methods on the two datasets, they cannot achieve good performance when searching over the TPS dataset, because of the limited size of relevant results in each user's specific dataset. Notice that the number of training samples in the TPS dataset is also much smaller than that in the Last.fm-1K dataset. On the TPS dataset, the absolute performance gain achieved by DL-MPTM over other methods for all the metrics are at least comparable to those in the Lastfm-1K dataset. This demonstrates a strong robustness of DL-MPTM on relatively small training datasets.

5.5.2 Effects of the Number of Latent Topics

In topic models, it is hard to accurately pre-define the number of topics, which has an important effect on the results. In the DL-MPTM model, there are two sets of latent topics: the number of latent music dimensions in the first layer, and the number of subtopics in the second layer. Fig. 5.3a and Fig. 5.3b

illustrate the effects of the two parameters, respectively. From the results, it can be observed that the number of latent music dimensions has strong impacts on the final performance, and it is optimal to set the number of music dimensions to [5, 20]. In contrast, we can observe the minor effects of sub-topic number, especially for Lastfm-1K dataset.

5.6 Summary

In this chapter, we present a personalized text-based music retrieval system which exploits the user listening behaviors in social music services. The system can accurately estimate the relevance of a song with respect to a term subject to user's music preference. To achieve the goal, a Dual-Layer Music Preference Topic Model is proposed to leverage the user listening logs and social tags to learn the interactions among (user, song, term), which are applied for personalized text-based music search. To evaluate the performance of the personalized retrieval system, comprehensive experiments have been conducted on two public datasets. The comparisons with the state-of-the-art text-based retrieval methods and existing personalized music retrieval methods in experiments show that our method can significantly improve the search performance in terms of accuracy. The results also demonstrate the importance of effective integration of personal music preference in developing high-performance music search engines, and verify the effectiveness of our proposed retrieval model.

Chapter 6

Conclusion

6.1 Thesis Summary

With personal mobile devices becoming the main music consuming platforms, there is an impending requirement on intelligent music information retrieval systems, which can provide personalized and context-aware music services. Music retrieval and music recommendation techniques are two main tools to help users find their favorite music. Text-based music retrieval is the most common paradigm for users to search music with semantic concepts, which could deliver user's current situation or music needs. However, no existing TBMR systems consider user's long-term music preferences. As a result, the search results cannot best satisfy different users. On the other hand, music recommendation could automatically recommend music which matches user's long-term music preferences, while ignoring the influences of local contexts on user's short-term music preferences. In recent years, the importance of local contexts on music preferences has been recognized and significant efforts have been invested into the development of context-aware music recommender (CAMR) systems. Although various CAMR systems have been developed, many problems still remain (see Section 2.2.2 in Chapter 2). In this thesis, we mainly focus on the development of personalized text-based music retrieval

and context-aware music recommender systems.

Venue plays an important role in music selection. For example, in restaurants or gyms, music is often used to create a suitable atmosphere. Besides, people also use music to facilitate their activities, such as meditation, running, reading or sleeping, which are also related to different venues. However, the effects of venue on music preferences have not been well explored in existing CAMR systems. Chapter 3 presents a venue-aware music recommender (VAMR) system for recommending suitable music to different types of venues. A latent topic model has been proposed to map both music and venues into a latent semantic space, in which they can be directly matched. Besides, in order to learn meaningful latent topics, a Music Concept Sequence Generation scheme is designed to represent each song as a sequence of music concepts. The VAMR system has been evaluated with *an offline experiments* on a small constructed dataset and a user study on a large scale music dataset.

In Chapter 4, we report a user information aware music retrieval system. The goal of the system is to leverage user demographic information (e.g., age and gender), which is easy to be obtained in mobile devices, to improve the search performance. Therefore, the system could be used for new users at the early stage in a personalized retrieval system to collect user's music listening behaviors. In this system, we proposed a user information aware music interest topic model to capture the influence of age and gender's music preferences. The music preferences related to age and gender are incorporated into a probabilistic retrieval method to exploit the influence of age and gender in text-based retrieval. An experimental study has been conducted and demonstrated that the effectiveness of the system.

Chapter 5 presents a personalized text-based music retrieval system. The relevance level of a song with respect to a query is subjective to different users. In other words, the preference levels of users on a song with respect to a query are different. Thus, it is crucial to capture the associations among

(user, song, term). To achieve the goal, we propose a dual-layer topic model: in the first layer, users music preferences are modeled with the latent topics discovered based on the co-occurrences of songs liked by different users; in the second layer, the latent topics are represented by the latent subtopics, which are discovered based on the co-occurrences of music contents in different songs. The correlations among user, song, and term can be well captured via the latent topics and subtopics. Then a probabilistic retrieval method is developed based on the dual-layer topic model. The system has been evaluated on two datasets and compared with various competitors. The experimental results show that the system can greatly improve the search accuracy with respect to user's music preference.

In summary, we develop a venue-aware music recommender system for CAMR and two text-based music retrieval systems for user-aware music retrieval. Through the development of these systems, main contributions we make include:

- To tackle the problem of "semantic gap" between high-level semantic concepts (used to describe music by human) and low-level audio features (used to represent music by computer), we use latent topic models to associate the semantic concepts and audio features via latent topics.

- The music preferences of users and contexts (i.e., venue) are modeled by the multinomial distributions of the latent topics. Besides, the characteristics of music tracks (i.e., songs) are also represented by the multinomial distributions of the latent topics. Therefore, in CAMR, the songs and contexts can be directly mapped in the latent space; in user-aware text-based music retrieval, the associations between user, song, and term (i.e., semantic concepts) are captured via the latent topics.

- For each system, we have constructed data collections and conducted experiments for evaluations. We hope that the empirical studies could

shed light on the methods of data construction and system evaluation for personalized and context-aware music retrieval and recommendation.

6.2 Future Work

In the development of personalized and context-aware music recommender systems, we identify several interesting and promising research problems in this direction for further exploration.

Construct a Comprehensive Musical Concept Vocabulary: A main challenge in music retrieval and recommendation is how to represent user's music preference based on the audio signal of music. People often describe music with semantic concepts. Thus, a general technique is to associate the audio features extracted from music signal with *music concepts* to bridge the "semantic gap". Here *music concepts* refer to the concepts used to describe music, such as era, mood, genre, instrument, etc. In Chapter 3, we use a small set of semantic concepts (i.e., mood, genre and instrument) in our experiments to demonstrate the effectiveness of our system. With more concepts being used, the characteristics of music can be described more comprehensively and precisely, which could improve the performance of the system. In Chapter 4 and Chapter 5, the social tags of songs in Last.fm are used. Social tags have a good coverage on the concepts used in music description, but they contain noisy terms. A good musical concept vocabulary is critical in modeling user's (contextual) music preference via the representation of semantic concepts. The important questions are: which concepts are useful and how many music concepts are sufficient. Thus, the construction of a comprehensive and concise music concept vocabulary is very useful for music retrieval and recommendation.

Construct Datasets for Personalized and Context-aware Music Retrieval and Recommendation: Datasets are critical for robust and reliable evaluation of information retrieval and recommender systems. In per-

sonalized music retrieval and context-aware music recommendation, there are no standard testbeds available, which makes the evaluation of related systems very difficult. In our experiments, we constructed several datasets by crawling multiple websites and through complex processing for system evaluation. The corresponding process is very time-consuming. As user-centric music retrieval and recommendation will be an important research direction in the future, it is important to develop standard datasets for evaluating and comparing different systems. Typically, there are two methods to construct such datasets: Web Mining and Crowdsourcing. Web mining is to crawl related data from multiple data sources, and then performs cross-platform matching to identify user's personal information or context information. For example, we can crawl the music-related tweets in Twitter, and then attempt to find out the places where users posted the tweets via Foursquare. In this way, the location context of music listening behaviors can be identified. Crowdsourcing is to rely on large scale users to collect related data explicitly or implicitly by designing webpages, games, and music applications (i.e., mobile music player) for users to label or listening to music tracks.

Exploit Song's Co-occurrence Patterns in Listening logs: Generally, a user likes different types of music/songs and the same song could be played by many times at different dates or under different situations. Users local music selection on songs is highly dependent on users local contexts, such as activity, mood, and surrounding environment. Under a certain context, users would like to listen to certain songs (a style of songs or a playlist of songs). The set of songs which a user prefers (or users personal music collection) will not be substituted/changed frequently. Instead, new songs will be gradually added to users personal music collection. Common users have regular daily routines, namely, a user will be often under several fixed contexts in his/her daily life. Therefore, the same set of songs will be played by many times under the same context at different dates. Consequently, there are co-occurrence patterns of

the songs played under different contexts. The songs could be organized into playlists based on their co-occurrence patterns, with the assumption that a playlist is suitable for a certain context for the user. Based on the patterns, users local music preferences could be inferred based on the songs they are listening to, which could be leveraged for music recommendation or improve the music search accuracy. For example, we could use the songs to which the user current listens to infer users current contexts or music preference and thus recommend suitable songs to the users or refine the search results.

Leverage the Correlations between User's Preferences on Other Media and User's Preferences on Music: In psychology and cognition studies, human perceptions of music and image are demonstrated to have a strong correlation, e.g., brain information processing of visual and audio are related [107] and music can stimulate of visual imagery [109]. In recent years, cross-modality modelling attracts lots of research attentions [59, 64, 166, 161]. Although most of those studies focus on the modelling of text and image, these techniques could be adapted to mine the relationship between text/image and music, e.g., mapping different modalities into a latent space and construct correlations among them in the latent space. There are also works modelling the relationship between image and music [145, 164] or text and music [24]. However, most of existing cross-modality modelling methods have not considered users personal preferences on documents (image, text, or music document). In our context, we could use personal preferences as a constraint in the construction of latent space for cross-modality mapping. Another problem is how to collect news/image and music data of users to train the model. In nowadays, a user usually has social accounts on multiple social platforms, such as Flickr (for image), Last.fm/Youtube (for music), Google news (for news). Thus, it is possible to collect user data from multiple datasets to construct training data. Based on the correlation between user's preferences on news/images and user's music preference, we could (1) infer users music preference based on their

news/image preferences; and (2) recommend suitable songs to users when users are reading news or browsing images based on the contents of news/songs.

Bibliography

[1] 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. Knowledge Data Eng.*, 17(6):734–749, 2005.

[2] P. Åman and L. A. Liikkanen. Painting the city with music: context-aware mobile services for urban environment. *Continuum*, pages 1–16, 2013.

[3] A. Ankolekar and T. Sandholm. Foxtrot: A soundtrack for where you are. In *Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications*, 2011.

[4] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on Context-Aware Recommender Systems*, 2009.

[5] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *Proceedings of the International Conference on Electronic Commerce and Web Technologies*, 2011.

[6] L. Baltrunas, M. Kaminskas, F. Ricci, L. Rokach, B. Shapira, and K.-H. Luke. Best usage context prediction for music tracks. In *Proceedings of the 2nd Workshop on Context Aware Recommender Systems*, 2010.

[7] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.

[8] L. Barrington, M. Yazdani, D. Turnbull, and G. R. Lanckriet. Combining feature kernels for semantic music retrieval. In *Proceedings of the International Society of Music Information Retrieval*, pages 614–619, 2008.

[9] V. Batagelj and M. Zaveršnik. Generalized cores. *arXiv preprint cs/0202039*, 2002.

[10] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, et al. Whozthat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network*, 22(4):50–55, 2008.

[11] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, 2007.

[12] P. Berkers. Gendered scrobbling: Listening behaviour of young adults on last.fm. *Interactions: Studies in Communication & Culture*, 2(3):279–296, 2012.

[13] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaumm. Hierarchical topic models and the nested chinese restaurant process. *Proceedings of Advances in Neural Information Processing Systems*, 16:17, 2004.

[14] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[16] D. Bogdanov, N. Wack, G. Emilia, G. Sankalp, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society of Music Information Retrieval*, 2013.

[17] F. Boström. Andromedia-towards a context-aware mobile music recommender. *Master thesis*, 2008.

[18] M. Braunhofer, M. Kaminskas, and F. Ricci. Recommending music for places of interest in a mobile travel guide. In *Proceedings of the ACM Conference on Recommender Systems*, 2011.

[19] M. Braunhofer, M. Kaminskas, and F. Ricci. Location-aware music recommendation. *International Journal of Multimedia Information Retrieval*, 2(1):31–44, 2013.

[20] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[21] J. C. Brown, O. Houix, and S. McAdams. Feature dependence in the automatic identification of musical woodwind instruments. *The Journal of the Acoustical Society of America*, 109(3):1064–1072, 2001.

[22] R. Burke. Hybrid web recommender systems. In P. Brusilovski, A. Kobsa, and W. Nejdl, editors, *The adaptive web*, chapter 12, pages 377–408. Springer, 2007.

[23] F. Cai, S. Liang, and M. de Rijke. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2014.

[24] R. Cai, C. Zhang, C. Wang, L. Zhang, and W.-Y. Ma. Musicsense: Contextual music recommendation using emotional allocation modeling. In *Proceedings of the ACM International Conference on Multimedia*, 2007.

[25] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing*, 41(3):271–284, 2005.

[26] P. Cano, M. Koppenberger, and N. Wack. Content-based music audio recommendation. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 211–212, 2005.

[27] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[28] Ò. Celma. Music recommendation. In Ò. Celma, editor, *Music Recommendation and Discovery*, chapter 3, pages 43–86. Springer, 2010.

[29] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27, 2011.

[30] C.-M. Chen, M.-F. Tsai, J.-Y. Liu, and Y.-H. Yang. Using emotional context from article for contextual music recommendation. In *Proceedings of the ACM international conference on Multimedia*, 2013.

[31] Z. Cheng and J. Shen. Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2014.

[32] Z. Cheng and J. Shen. Venuemusic: A venue-aware music recommender system. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

[33] Z. Cheng and J. Shen. Venuemusic: A venue-aware music recommender system. *IEEE Trans. Inf. Syst.*, xx(x), 2016.

[34] S. Cunningham, S. Caulder, and V. Grout. Saturday night or fever? context-aware music playlists. *Proc. Audio Mostly*, 2008.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[36] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. 2011.

[37] J. Donaldson. A hybrid social-acoustic recommendation system for popular music. In *Proceedings of the ACM Conference on Recommender systems*, 2007.

[38] S. Dornbush, J. English, T. Oates, Z. Segall, and A. Joshi. Xpod: A human activity aware learning mobile music player. In *Proceedings of the IJCAI Workshop on Ambient Intelligence*, 2007.

[39] J. S. Downie. Mirex 2014 evaluation results, 2014.

[40] G. T. Elliott and B. Tomlinson. Personalsoundtrack: context-aware playlists that adapt to user pace. In *CHI'06 extended abstracts on Human factors in computing systems*, 2006.

[41] K. Ellis, E. Coviello, A. B. Chan, and G. R. Lanckriet. A bag of systems representation for music auto-tagging. *IEEE Trans. Audio, Speech, and Language Process.*, 21(12):2554–2569, 2013.

[42] S. Essid, G. Richard, and B. David. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Trans. Audio, Speech, Language Process.*, 14(1):68–80, 2006.

[43] F. Eyben, M. Wöllmer, and B. Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the ACM International Conference on Multimedia*, 2010.

[44] L. Gaye, R. Mazé, and L. E. Holmquist. Sonic city: The urban environment as a musical interface. In *Proceedings of the Conference on New Interfaces for Musical Expression*, 2003.

[45] A. E. Greasley and A. M. Lamont. Music preference in adulthood: Why do we like the music we do. In *Proceedings of the International Conference on Music Perception and Cognition*, 2006.

[46] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

[47] A. Hannak, P. Sapiezynski, A. M. Kakhki, B. Krishnamurth, D. Lazer, A. Mislove, and C. Wilson. Measuring personalization of web search. In *Proceedings of the International Conference Companion on World Wide Web*, 2013.

[48] N. Hariri, B. Mobasher, and R. Burke. Personalized text-based music retrieval. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[49] G. Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[50] Ò. C. Herrada. Music recommendation and discovery in the long tail. *PhD Thesis*, 2009.

[51] K. Hoashi, K. Matsumoto, and N. Inoue. Personalization of user profiles for content-based music retrieval based on relevance feedback. In *Proceedings of the ACM International Conference on Multimedia*, 2003.

[52] M. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

[53] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[54] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proceedings of International Joint Conferences on Artificial Intelligence*, 1999.

[55] M. B. Holbrook and R. M. Schindler. Some exploratory findings on the development of musical tastes. *Journal of Consumer Research*, 16(1):119–124, 1989.

[56] P. Hu, W. Liu, W. Jiang, and Z. Yang. Latent topic model for audio retrieval. *Pattern Recognition*, 47(3):1138–1143, 2014.

[57] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[58] J. H. Jensen, D. P. Ellis, M. G. Christensen, and S. H. Jensen. Evaluation distance measures between gaussian mixture models of mfccs. In *Proceedings of the International Conference on Music Information Retrieval*, 2007.

[59] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. In *IEEE International Conference on Computer Vision*, 2011.

[60] B. jun Han, S. Rho, S. Jun, and E. Hwang. Music emotion classification and context-based music recommendation. *Multimed. Tools Appl.*, 47(3):433–460, 2010.

[61] M. Kaminskas, I. Fernández-Tobías, F. Ricci, and I. Cantador. Knowledge-based music retrieval for places of interest. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 19–24, 2012.

[62] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In *Proceedings of the International Conference on User Modeling, Adaption, and Personalization*, 2011.

[63] M. Kaminskas, F. Ricci, and M. Schedl. Location-aware music recommendation using auto-tagging and hybrid matching. In *Proceedings of the ACM Conference on Recommender Systems*, 2013.

[64] C. Kang, S. Liao, Y. He, J. Wang, W. Niu, S. Xiang, and C. Pan. Cross-modal similarity learning: A low rank bilinear formulation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.

[65] B. F. Karlsson, K. Okada, and T. Noleto. A mobile-based system for context-aware music recommendations. In *Artificial Intelligence Applications and Innovations*, pages 520–529. Springer, 2012.

[66] D.-k. Kim, G. Voelker, and L. K. Saul. A variational approximation for topic modeling of hierarchical corpora. In *Proceedings of the Annual International Conference on Machine Learning*, 2013.

[67] J.-H. Kim, C.-W. Song, K.-W. Lim, and J.-H. Lee. Design of music recommendation system using context information. In *Agent Computing and Multi-Agent Systems*, pages 708–713. Springer, 2006.

[68] J.-Y. Kim and N. J. Belkin. Categories of music description and search terms and phrases used by non-music experts. In *Proceedings of the International Society of Music Information Retrieval*, 2002.

[69] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, and K. Seyerlehner. A document-centered approach to a natural language music search engine. In *Proceedings of European Conference on Information Retrieval*, 2008.

[70] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner, and G. Widmer. Augmenting text-based music retrieval with audio similarity. In *Proceedings of the International Society of Music Information Retrieval*, 2009.

[71] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[72] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[73] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[74] M. Kuhn, R. Wattenhofer, and S. Welten. Social audio features for advanced music retrieval interfaces. In *Proceedings of the International Conference on Multimedia*, 2010.

[75] F.-F. Kuo, M.-F. Chiang, M.-K. Shan, and S.-Y. Lee. Emotion-based music recommendation by association discovery from film music. In *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005.

[76] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008.

[77] A. Lamont and A. Greasley. Chapter 15: Music preferences. In S. Hallam, I. Cross, and M. Thaut, editors, *Oxford Handbook of Music Psychology*. Oxford University Press, 2009.

[78] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[79] O. Lartillot and P. Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International Conference on Digital Audio Effects*, 2007.

[80] A. LeBlanc, J. Colman, J. McCrary, C. Sherrill, and S. Malin. Tempo preferences of different age music listeners. *Journal of research in music education*, 36(3):156–168, 1988.

[81] A. LeBlanc, Y. Jin, L. Stamou, and J. McCrary. Effect of age, country, and gender on music listening preferences. *Bulletin of the Council for Research in Music Education*, (141):72–76, 1999.

[82] J. S. Lee and J. C. Lee. Music for my mood: A music recommendation system based on context reasoning. In *Proceedings of 1st European Conference on Smart Sensing and Context*, 2006.

[83] J. S. Lee and J. C. Lee. Context awareness by case-based reasoning in a music recommendation system. In *Ubiquitous Computing Systems*, pages 45–58. Springer, 2007.

[84] A. Lehtiniemi. Evaluating supermusic: streaming context-aware mobile music service. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, 2008.

[85] D. J. Levitin and J. McGill. Life soundtracks: The uses of music in everyday life. Technical report, 2007.

[86] M. Levy and M. Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137–150, 2008.

[87] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *IEEE Trans. Multimed.*, 11(3):383–395, 2009.

[88] C.-T. Li and M.-K. Shan. Emotion-based impressionism slideshow with automatic music accompaniment. In *Proceedings of the 15th international conference on Multimedia*, 2007.

[89] C. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic. The need for music information retrieval with user-centered and multimodal strategies. In *Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2011.

[90] H. Liu, J. Hu, and M. Rauterberg. Music playlist recommendation based on user heartbeat and music preference. In *Proceedings of the International Conference on Computer Technology and Development*, 2009.

[91] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Society of Music Information Retrieval*, 2000.

[92] L. Lu, H. Jiang, and H. Zhang. A robust audio classification and segmentation method. In *Proceedings of the ACM International Conference on Multimedia*, 2001.

[93] L. Lu, D. Liu, and H.-J. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Trans. Audio, Speech, Language Process.*, 14(1):5–18, 2006.

[94] A. Majumder and N. Shrivastava. Know your personalization: learning topic level personalization in online services. In *Proceedings of the International Conference Companion on World Wide Web*, 2013.

[95] X.-L. Mao, Z.-Y. Ming, T.-S. Chua, S. Li, H. Yan, and X. Li. SSHLDA: a semi-supervised hierarchical topic model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2012.

[96] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard. Yaafe, an easy to use and efficient audio feature extraction software. In *Proceedings of the International Society of Music Information Retrieval*, 2010.

[97] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. The million song dataset challenge. In *Proceedings of the International Conference Companion on World Wide Web*, 2012.

[98] S. Miller, P. Reimer, S. R. Ness, and G. Tzanetakis. Geoshuffle: Location-aware, content-based music browsing using self-organizing rag clouds. In *Proceedings of the International Society of Music Information Retrieval*, 2010.

[99] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2002.

[100] R. Miotto and G. Lanckriet. A generative context model for semantic music annotation and retrieval. *IEEE Trans. Audio, Speech, and Language Process.*, 20(4):1096–1108, 2012.

[101] R. Miotto and N. Orio. A probabilistic model to combine tags and acoustic similarity for music retrieval. *ACM Trans. Inf. Syst.*, 30(2):8, 2012.

[102] J. Nam, J. Herrera, M. Slaney, and J. O. Smith. Learning sparse feature representations for music annotation and retrieval. In *Proceedings of the International Society of Music Information Retrieval*, pages 565–570, 2012.

[103] H. Y. nd Bin Cui, Y. Sun, Z. Hu, and L. Chen. Lcars: A spatial item recommender system. *ACM Trans. Inf. Syst.*, 32(3):11, 2014.

[104] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012.

[105] A. C. North, D. J. Hargreaves, and J. J. Hargreaves. Uses of music in everyday life. *Music perception*, 22(1):41–77, 2004.

[106] A. C. North, D. J. Hargreaves, and J. J. Hargreaves. Uses of music in everyday life. *Music Perception*, 22(1):41–77, 2004.

[107] I. R. Olson, J. C. Gatenby, and J. C. Gore. A comparison of bound and unbound audio–visual information processing in the human cerebral cortex. *Cognitive Brain Research*, 14(1):129–138, 2002.

[108] N. Orio. *Music retrieval: A tutorial and review.* now publishers Inc, 2006.

[109] J. W. Osborne. The mapping of thoughts, emotions, sensations, and images as responses to music. *Journal of Mental Imagery*, 5(5):133–136, 1981.

[110] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):256–270, 2015.

[111] B. Pardo, J. Shifrin, and W. Birmingham. Name that tune: A pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology*, 55(4):283–300, 2004.

[112] H.-S. Park, J.-O. Yoo, and S.-B. Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *Proceedings of the 3rd International Conference on Fuzzy Systems and Knowledge Discovery*, 2006.

[113] A. J. Perotte, F. Wood, N. Elhadad, and N. Bartlett. Hierarchically supervised latent dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2011.

[114] Y. Petinot, K. McKeown, and K. Thadani. A hierarchical model of web summaries. In *The Annual Meeting of the Association for Computational Linguistics*, 2011.

[115] T. F. Pettijohn II, G. M. Williams, and T. C. Carter. Music for the seasons: seasonal music preferences in college students. *Current Psychology*, 29(4):328–345, 2010.

[116] A. Popescul, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2001.

[117] D. Putthividhy, H. T. Attias, and S. S. Nagarajan. Topic regression multimodal latent dirichlet allocation for image annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[118] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2009.

[119] S. Reddy and J. Mascia. Lifetrak: Music in tune with your life. In *Proceedings of the ACM International Workshop on Human-Centered Multimedia*, 2006.

[120] P. J. Rentfrow and S. D. Gosling. The do re mi's of everyday life: The structure and personality correlates of music preferences. *J. Pers. Soc. Psychol.*, 84(6):1236–1256, 2003.

[121] Z. Resa. Towards time-aware contextual music recommendation: an exploration of temporal patterns of music listening using circular statistics. *Master thesis*, 2010.

[122] 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*, 1994.

[123] S. Rho, B. jun Han, and E. Hwang. Svr-based music mood classification and context-based music recommendation. In *Proceedings of the ACM International Conference on Multimedia*, 2009.

[124] F. Ricci. Context-aware music recommender systems: workshop keynote abstract. In *Proceedings of the International Conference Companion on World Wide Web*, 2012.

[125] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[126] M. K. F. Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2):89–119, 2012.

[127] M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008.

[128] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers. Learning author-topic models from text corpora. *ACM Trans. Inf. Syst.*, 28(1):4, 2010.

[129] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[130] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000.

[131] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovski, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, chapter 9, pages 291–324. Springer, 2007.

[132] M. Schedl, G. Breitschopf, and B. Ionescu. Mobile music genius: Reggae at the beach, metal on a friday night? In *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2014.

[133] M. Schedl and A. Flexer. Putting the user in the center of music information retrieval. In *Proceedings of the International Society of Music Information Retrieval*, 2012.

[134] M. Schedl, E. Gómez, and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends® in Information Retrieval*, 8(2-3):127–261, 2014.

[135] M. Schedl and D. Schnitzer. Location-aware music artist recommendation. In *Proceedings of the International Conference on MultiMedia Modeling*, 2014.

[136] M. Schedl, S. Stober, E. Gómez, N. Orio, and C. C. Liem. User-aware music retrieval. *Multimodal Music Processing*, 3:135–156, 2012.

[137] C. Schörkhuber and A. Klapuri. Constant-q transform toolbox for music processing. In *Proceedings of the 7th Sound and Music Computing Conference*, 2010.

[138] J. Seppänen and J. Huopaniemi. Interactive and context-aware mobile music experiences. In *Proceedings of the 11th Int. Conference on Digital Audio Effects*, 2008.

[139] U. Shardanand and P. Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

[140] J. Shen, J. Shepherd, and A. H. Ngu. Towards effective content-based music retrieval with multiple acoustic feature combination. *IEEE Trans. Multimed.*, 8(6):1179–1189, 2006.

[141] A. Singla, R. W. White, A. Hassan, and E. Horvitz. Enhancing personalization via search activity attribution. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2014.

[142] Y. Song, S. Dixon, and M. Pearce. A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*, 2012.

[143] D. N. Sotiropoulos, A. S. Lampropoulos, and G. A. Tsihrintzis. Musiper: a system for modeling music similarity perception based on objective feature subset selection. *User Modeling and User-Adapted Interaction*, 18(4):315–348, 2008.

[144] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proc. of ACM KDD*, 2004.

[145] A. Stupar and S. Michel. Picasso - to sing, you must close your eyes and draw. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[146] J.-H. Su, H.-H. Yeh, P. S. Yu, and V. S. Tseng. Music recommendation using content and context information mining. *IEEE Intell. Syst.*, 25(1):16–26, 2010.

[147] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang. A systematic evaluation of the bag-of-frames representation for music information retrieval. *IEEE Trans. Multimed.*, 16(5):1188–1200, 2014.

[148] P. Symeonidis, M. M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos. Ternary semantic analysis of social tags for personalized music recommendation. In *Proceedings of the International Society of Music Information Retrieval*, 2008.

[149] J. Tang, S. We, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[150] D. Tingle, Y. E. Kim, and D. Turnbull. Exploring automatic music annotation with acoustically-objective tags. In *Proceedings of the International Conference on Multimedia Information Retrieval*, 2010.

[151] D. Turnbull, L. Barrington, D. Torres, and G. R. Lanckriet. Towards musical query-by-semantic-description using the cal500 data set. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[152] D. R. Turnbull, L. Barrington, G. Lanckriet, and M. Yazdani. Combining audio content and social context for semantic music discovery. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.

[153] R. Typke, F. Wiering, and R. C. Veltkamp. A survey of music information retrieval systems. In *Proceedings of the International Society of Music Information Retrieval*, 2005.

[154] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.*, 10(5):293–302, 2002.

[155] S. Virtanen, Y. Jia, A. Klami, and T. Darrell. Factorized multi-modal topic model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2012.

[156] A. Wang. The shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006.

[157] C. Wang, J. Wang, X. Xie, and W.-Y. Ma. Mining geographic knowledge using location aware topic model. In *Proceedings of the ACM workshop on Geographical Information Retrieval*, 2007.

[158] J.-C. Wang, Y.-C. Shih, M.-S. Wu, H.-M. Wang, and S.-K. Jeng. Colorizing tags in tag cloud: a novel query-by-tag music search system. In *Proceedings of the ACM International Conference on Multimedia*, 2011.

[159] X. Wang, D. Rosenblum, and Y. Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the ACM International Conference on Multimedia*, 2012.

[160] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang. PLDA: Parallel latent dirichlet allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer, 2009.

[161] Y. Wang, F. Wu, J. Song, X. Li, and Y. Zhuang. Multi-modal mutual topic reinforce modeling for cross-media retrieval. In *Proceedings of the ACM International Conference on Multimedia*, 2014.

[162] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the International Conference Companion on World Wide Web*, 2013.

[163] G. Wijnalda, S. Pauws, F. Vignoli, and H. Stuckenschmidt. A personalized music system for motivation in sport performance. *IEEE pervasive computing*, 4(3):26–32, 2005.

[164] X. Wu, Y. Qiao, X. Wang, and X. Tang. Briding music and image via cross-modal ranking analysis. *IEEE Transcations on Multimedia*, pp(99):xxx–yyy, 2016.

[165] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Trans. Audio, Speech, and Language Process.*, 16(2):435–447, 2008.

[166] Q. You, J. Luo, H. Jin, and J. Yang. Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, 2016.

[167] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma. LightLDA: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, 2015.

[168] K. Zhai and J. Boyd-Graber. Online latent Dirichlet allocation with infinite vocabulary. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[169] B. Zhang, J. Shen, Q. Xiang, and Y. Wang. Compositemap: A novel framework for music similarity measure. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.

[170] B. Zhang, Q. Xiang, H. Lu, J. Shen, and Y. Wang. Comprehensive query-dependent fusion using regression-on-folksonomies: A case study of multi-modal music search. In *Proceedings of the ACM International Conference on Multimedia*, 2009.

Appendix A

Evaluation Metrics

This section introduces the evaluation metrics used in the dissertation.

Precision is the fraction of the documents retrieved that are relevant to the query.

Precision@k (P@k) is the proportion of relevant documents in the top k results, computed as:

$$Precision@k = \frac{\text{No. of relevant items in top } k \text{ results}}{k} \tag{A.1}$$

Average Precision (AP) averages the precision at each point of a relevant songs in the ranking list. It measures the quality of the whole ranking list.

$$AP@k = \frac{\sum_i^k Precision@i \cdot \delta(rel_i = 1)}{min(k, |rel|)} \tag{A.2}$$

where rel_i indicates the relevance of the i-th song in the ranking list. If the i-th song is relevant, $rel_i = 1$; otherwise, $rel_i = 0$. $\delta(\cdot)$ is a binary indicator function. $|rel|$ is the number of relevant songs in the dataset.

Mean Average Precision (MAP) is the mean of the average precision scores for a set of queries.

Normalized Discounted Cumulative Gain (NDCG) [57] uses a graded relevance scale of documents from the result set to evaluate the usefulness of a document based on its position in the result lists. NDCG@k is widely used for

measuring the rank accuracy, defined as

$$NDCG@k = \frac{1}{Z_k} \sum_{j=1}^{k} \frac{2^{r(j)} - 1}{log_2(j+1)} \tag{A.3}$$

where j is the rank position, $r(j)$ is the rating value of j-th song in the ground-truth rank list, Z_k is the normalization factor which is the discounted cumulative gain in the k-th position of the ground truth rank list. In the computation of NDCG@k, the rating value of relevant, neutral and irrelevant items are 2, 1, and 0, respectively.

MRR averages the inverse of the rank of the first correct answer for each query. It measures the level of the ranking list at which the information need of the user is first fulfilled.