3-7-2011

User-centric Music Information Retrieval

Bo Shao
Florida International University, bo.shao@sunguide.org

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

USER-CENTRIC MUSIC INFORMATION RETRIEVAL

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Bo Shao

2011

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Bo Shao, and entitled User-Centric Music Information Retrieval, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Shu-Ching Chen

Vagelis Hristidis

Mohammed Hadi

Tao Li, Major Professor

Date of Defense: March 7, 2011

The dissertation of Bo Shao is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Interim Dean Kevin O'Shea
University Graduate School

Florida International University, 2011

DEDICATION

I dedicate this dissertation to my mother. Without her patience, understanding, support, and most of all love, the completion of this work would not have been possible.

ACKNOWLEDGMENTS

I would like to deeply thank to my advisor, Dr. Tao Li, for guiding me in my research. In the past five years, he has provided me insightful advices and patient guidance. He has also given me valuable advices on career development.

I want to thank Professor Shu-Ching Chen, Professor Vagelis Hristidis, and Professor Mohammed Hadi for serving on my committee. They have provided me many constructive questions and useful suggestions.

I extend my warmest thanks to Mr. Charles Robbins (P.E.), my manager at AECOM Corporation for his support and encouragement of my Ph.D study and research.

The School of Computer Science at the Florida International University is a great place for study and research. I enjoyed my life here with all the faculty members, department staff, and fellow graduates. I am grateful for their support and friendship. I want to specially thank my lab-mates: Wei Peng, Dingding Wang, Yi Zhang, Fei Wang, Xin Wang, Lei Li and Jingxuan Li.

I would like to express my special acknowledgment to my wife, Wei Guo, for her unfailing and unconditional support, and to my son, Kenny Shao, for his love and understanding.

The material presented in this thesis is based upon work that is partially supported by NSF under IIS-0546280, HRD-0317692, and IIP-0450552. Any opinions, findings, conclusions or recommendations expressed in this material are those of myself and my coauthors and do not necessarily reflect the views of the funding organizations.

ABSTRACT OF THE DISSERTATION

USER-CENTRIC MUSIC INFORMATION RETRIEVAL

by

Bo Shao

Florida International University, 2011

Miami, Florida

Professor Tao Li, Major Professor

The rapid growth of the Internet and the advancements of the Web technologies have made it possible for users to have access to large amounts of on-line music data, including music acoustic signals, lyrics, style/mood labels, and user-assigned tags. The progress has made music listening more fun, but has raised an issue of how to organize this data, and more generally, how computer programs can assist users in their music experience.

An important subject in computer-aided music listening is music retrieval, i.e., the issue of efficiently helping users in locating the music they are looking for. Traditionally, songs were organized in a hierarchical structure such as genre->artist->album->track, to facilitate the users' navigation. However, the intentions of the users are often hard to be captured in such a simply organized structure. The users may want to listen to music of a particular mood, style or topic; and/or any songs similar to some given music samples. This motivated us to work on user-centric music retrieval system to improve users' satisfaction with the system.

The traditional music information retrieval research was mainly concerned with classification, clustering, identification, and similarity search of acoustic data of music by way of feature extraction algorithms and machine learning techniques. More recently the music information retrieval research has focused on utilizing other types of data, such as lyrics, user-access patterns, and user-defined tags, and on targeting non-genre categories for classification, such as mood labels and styles. This dissertation focused on investigating and developing effective data mining techniques for (1) organizing and annotating music data with styles, moods and user-assigned tags; (2) performing effective analysis of music data with features from diverse information sources; and (3) recommending music songs to the users utilizing both content features and user access patterns.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Music is very popular in modern life, and the amount of digital music available to music listeners has increased dramatically. In computer science, researchers have been intensely working on developing techniques for computationally dealing with music data. In particular, the development of efficient and effective computational assistants in music listening has recently become more and more urgent due to the high demand from web-based music stores and services.

An important subject in computer-aided music listening is music retrieval, i.e., the issue of efficiently helping users in locating the music they are looking for. Traditionally, songs were organized in a hierarchical structure such as genre->artist->album->track, to facilitate the users' navigation. Some websites or systems allow users to create their own playlists so that songs can be organized into a preferred personal collection. However, the intentions of the users are often hard to be captured in such simply organized structures. The users may want to listen to music of a particular mood, style or topic; and/or any songs similar to some given music samples. This motivates us to work on user-centric music retrieval system to improve users' satisfaction with the system.

In particular, the goal of this research is to investigate and develop data mining techniques to create a practical system that allows users to effectively and efficiently retrieve music. More specifically, there are three closely related dimensions of this research theme:

- **music data organization and annotation:** How can we organize and annotate music by appropriate labels, not only by artists, album titles, track titles, and genres, but also by styles, moods and user-assign tags?

- **music analysis from different information sources:** Given the music data that are often represented by multiple sets of features (e.g., audio content, meta-data, lyrics etc.), how can we perform effective music analysis from these diverse information sources?

- **music recommendation:** How can we develop good music recommendation systems based on a good understanding of the users' preferences and the music pieces in the collection?

1.2 Background

In the past decade, music information retrieval has been receiving a considerable amount of attention, e.g. [2, 44], but the state-of-the-art music retrieval techniques are still far from mature and often fail to deliver satisfactory results. Various music retrieval approaches have been developed, and music meta data, content data, user listening history have been utilized for these approaches to work. Multimedia conferences, e.g. ISMIR (International Conference on Music Information Retrieval) and WEDELMUSIC (Web Delivering of Music), have a focus on the development of computational techniques for indexing, classifying, summarizing and analyzing music data. Most of the previous researches on music retrieval focused on music representation and its use in similarity search [8,30,47]. More details of the background review can be found in the chapter 2.

1.2.1 Music Organization and Annotation

Huron [61] points out that, because the preeminent functions of music are social and psychological, the most useful characterization of music would be based on a variety of information including genre, style, mood, and similarity. Therefore, to enable music queries, it is imperative that each piece of music be annotated by appropriate labels, not only by artists, album titles, track titles, and genres, which in many cases are readily available at *GraceNote* (`http://www.gracenote.com`), an access-free on-line database, but also by more pertinent information, such as style and mood labels, which are available at online music stores and *AllMusic* (`http://www.allmusic.com`), a registered-user-only on-line database. Music annotation considers the problem of automatically assigning the latter type of labels so as to eliminate the need of accessing those limited-access databases. Recently, the user-assigned tags have turned into an essential component in music information retrieval and the problem of automatic music annotation using user-assigned tags has also attracted a lot of research attention.

1.2.2 Music Analysis from Different Information Sources

In music information retrieval, the data are naturally multi-modal, in the sense that they are represented by multiple sets of features. For example, the representation of a song has four modes: 1) the personnel (the producer, the director, the editor, the scenario writer, the music composer, the cast, etc.), 2) the lyric features, 3) the user-assigned tags, and 4) the acoustic features (which summarize the voice and the background audio). Having data with heterogeneous sets of features, one may pose a natural question: can multi-modality be effectively utilized in music data analysis, and if so, can such multi-modal learning methods produce better analysis results than uni-modal methods?

1.2.3 Music Recommendation

Music recommendation is an important component of music information retrieval. The goal of music recommendation is to present users lists of songs that they are likely to enjoy. Music recommendation should be based on a good understanding of the users' preferences and the music pieces in the collection. Collaborative-filtering and content-based recommendations are two approaches that have been widely used for this purpose. However, both approaches have their own disadvantages: collaborative-filtering methods need a large collection of user history data and content-based methods lack the ability of understanding the interests and preferences of users. Therefore, new techniques are needed for effective music recommendation.

1.3 Contribution of this Dissertation

The traditional music information retrieval research was mainly concerned with classification, clustering, identification, and similarity search of acoustic data of music by way of feature extraction algorithms and machine learning techniques. More recently the music information retrieval research has focused on utilizing other types of data, such as lyrics, user-access patterns, and user-defined tags, and on targeting non-genre categories for classification, such as mood labels and styles. My dissertation focuses on investigating and developing effective data mining techniques for (1) organizing and annotating music data with styles, moods and user-assigned tags; (2) performing effective analysis of music data with features from diverse information sources; and (3) recommending music songs to the users utilizing both content features and user access patterns. The main contribution of my work can be summarized as follows:

1.3.1 Music Organization and Annotation

Music organization and annotation is the foundation of an intelligent music retrieval system. More and more social-networking music listening websites are providing user-defined tags, styles, and mood labels to help users to make quick selections of music songs. In this dissertation, we develop new techniques to correlate style and mood models and also perform multi-label mood/style classification by making use of user-assigned tags.

Correlating styles and mood labels [134]: An important characteristic of the style and mood labels is that most labels are having close semantic relationships. The first type of the relationships is that some labels are synonyms, e.g., “witty” and “thoughtful”, “happy” and “cheerful”. The second type of the relationships is that some labels are more general while some others are more specific, e.g., “Soft Metal” is a more specific style than “Metal”, “Dance Pop” is a more specific style than “Pop”, “Extremely Provocative” is a more specific tag than “Provocative”, and “Agony” is a more specific mood label than “Sadness”. One challenge is whether we can automatically characterize such semantic relations among the labels using a hierarchical structure. In this dissertation, we develop a hierarchical divisive co-clustering algorithm for exploring the relationships among the style/mood models. The discovered relationship can be used to compute the similarity between music artists.

Multi-label mood/style classification [150]: Traditional music mood/style classification approaches assumed that each piece of music had a unique mood/style and they made use of the music content (audio features) to construct a classifier for classifying each piece into its unique mood/style. However, in reality, a piece of music may match more than one, even several different moods/styles. In addition, how to incorporate the tag information into the classification process is also a challenge. In this dissertation, we develop a novel multi-label music mood/style classification approach with hypergraph regularization. The proposed approach also integrates both music content and user-assigned tags for classification.

1.3.2 Music Analysis from Different Information Sources

In music information retrieval, the data are naturally multi-modal, in the sense that they are represented by multiple sets of features. In this dissertation, we study the issue of clustering pop music into groups with respect to the artists from diverse information sources. In particular, we develop algorithms to improve the performance of clustering by integrating different information sources [86].

1.3.3 Music Recommendation

Music recommendation aims to provide a music listener a list of music pieces that he/she is likely to enjoy. It needs to satisfy the following two requirements [147]: (1) High recommendation accuracy. A good recommendation system should output a relatively short list of songs in which many pieces are favored and few pieces are not favored; (2) High recommendation novelty. Good novelty is defined as rich artist variety / diversity and well-balanced music content variety / diversity. Therefore, effective music recommendation should be based on a good understanding of the preferences of the users and the music pieces in the collection. The key to a success music recommendation system is to develop a good measurement strategy of the music similarity and an effective recommendation method based on the similarity measurement. In this dissertation, we develop a music recommendation approach by incorporating collaborative-filtering approach and acoustic contents of music. The new approach employs a novel dynamic music similarity measurement strategy, which significantly improves the similarity measurement in terms of accuracy and efficiency. This measurement strategy utilizes the user access patterns from large numbers of users and represents music similarity with an undirected graph. Recommendation is then calculated using the graph Laplacian and label propagation defined over the graph [135].

1.3.4 System Development and Evaluation

We also develop a prototype system for multi-modal music information retrieval and a real world user-centric music retrieval web application for evaluating our proposed techniques.

Please note that although the proposed algorithms and approaches are only evaluated based on the music data and applied exclusively in the music information retrieval research area, many of them can be adopted or at least adapted to handle other types of data and address the problems in other research areas.

1.4 Dissertation Outline

The rest of the Dissertation is organized as follows: Chapter 2 provides the literature review. Chapter 3 introduces our proposed techniques for exploring the relationships among the style/mood models and for multi-label style/mood classification. Chapter 4 studies the problem of identifying “similar” artists using features from diverse information sources and presents the clustering algorithms that integrate features from both music content and lyrics to perform bimodal learning. Chapter 5 discusses our proposed approach for music recommendation by incorporating collaborative-filtering and acoustic contents of music. Chapter 6 describes our developed prototype system and the real world user-centric music retrieval web application. Finally, Chapter 7 concludes the dissertation and discusses future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Efficient and intelligent Music Information Retrieval (MIR) is a very important topic of the 21st century. With the ultimate goal of building user-centric music information retrieval systems, this chapter studies the problems of existing MIR approaches and systems. We will first review different music data sources, and try to associate various music retrieval tasks to each type of the data. This will answer the question of what types of music data are available and for each data type, what retrieval tasks are often performed in the literature. We will then study different approaches used for each task, attempting to answer the question of what data mining algorithms or techniques are mostly used for each task.

2.2 Music Data Types and Associated Retrieval Tasks

Table 1 illustrates different music data types [79] used for music information retrieval tasks. As suggested in [76], these data types are grouped into two categories in the table: 1) music content data; and 2) social context data. In order to better understand the nature of each type of the music data sources, we will discuss each of them and review various music retrieval tasks that are based on analyzing these two different categories of data.

2.2.1 Content Features and Content-based Music Retrieval

Based on the content features used, content-based music retrieval methods can be categorized into three major branches as discussed in the following subsections.

Music Retrieval Based on Acoustic Features

Acoustic features are exacted from audio data, which are audio recordings in a format like WAV, AIFF, AU, MP3 or WMA. They are the essential part of the music objects for music listeners as well as the core in music retrieval systems. MIR research community has been

Data Category	Data Type	Details or Examples
Music Content Data	Music Metadata	artist name, track title, track duration, album, publisher, publishing date
	Lyrics	lyrics in text
	Audio Data	audio recordings in format like WAV, AIFF, AU, MP3, WMA
	Symbolic Data	MIDI files, MusicXML, Humdrum
	Music Scores	music score notations
Social Context Data	Expert Annotations	genre, mood, style
	Music Reviews	comments or feedback of music listeners, generally in a very long loose description
	Social Tags	comments or feedback of music listeners, generally in a concise textual format
	User Profiles	created by music listeners on certain music websites to record the user preferences of music
	Playlists	list of songs that music listeners created on music websites or for personal music collections

Table 1: Various music data sources

focusing on this data, trying to extract various types of acoustic features for different purposes, and making use of them in different tasks and applications. As most of the work in the MIR research area, of the proposed techniques, and of the developed systems are mainly based on acoustic features or partially utilize acoustic features, it is critical for us to understand the most commonly used features [107]:

- Pitch: It represents the perceived fundamental frequency of a sound. Pitch can range from low or deep to high or acute. It is one of the major auditory attribute of the musical tones. Note that pitch is related to frequency, but they are not equivalent. Frequency is the scientific measure of pitch and is objective, while pitch is completely subjective.

- Intensity / loudness: It defines the amplitude of the sound vibration, the primary psychological correlate of physical strength. It ranges from soft to loud. It is another major auditory attribute of the music tones.

- Timbre: It is defined as the sound characteristics that allow listeners to distinguish sounds even when they have the same pitch and intensity (loudness). Music listeners are generally sensitive to the timbre feature. For example, a piano generates sounds with a very different timbre from the sounds generated by a violin. A music fan might favor one singer's songs mainly because of the unique timbre features of the sound produced by the singer. In MIR research community, it is frequently used as one of the main features for music genre classification.

- Tempo: It is the speed at which a musical work is played, or expected to be played, by performers. Tempo is usually measured in beats per minute (bpm).

- Orchestration: It is the study or practice for a music ensemble or of adapting for orchestra music composed for another medium. It is often based on the choice of the composers and the performers in selecting which musical instruments are to be employed to play the different voices, chords, and percussive sounds of a musical work. The orchestration decision can sometime dramatically affect the music style. For example, one selection of orchestration can make one music piece dignified, while another selection can make the same music played funny or cheerful.

- Rhythm: It is related to the periodic repetition, with possible small variants, of a temporal pattern of onsets alone. In other words, rhythm is the timing pattern of the musical sounds and silences. Different rhythms can be perceived by the listeners at the same time in the case of polyrhythmic music.

- Melody: It is a sequence of musical tones which is perceived as a single entry. The tones in a melody generally have a similar timbre with a recognizable pitch in a small frequency range. Melodies often consist of one or more musical phrases or motifs, and are usually repeated throughout a music piece in various forms. Note that melody is used in different ways in different music styles, and in consequence, it is an important acoustic feature for MIR research.

- Harmony: It is the organization of simultaneous sounds with a recognizable pitch along the time axis. Harmony refers to the “vertical” aspect of music, as distinguished from melodic line, which is the “horizontal” aspect.

In literature, different sets of acoustic features are extracted for different applications. The popular feature sets are summarized as follows:

- Timbral texture feature set, rhythmic content feature set, and pitch content features set: Proposed in [146] and implemented by George Tzanetakis, they are widely adopted for the applications and studies of music genre classification and music recommendation, including many of our research studies presented in this dissertation.

- Standard Low-Level (SLL) signal parameters, MFCC, Psychoacoustic (PA) features, Auditory Filterbank Temporal Envelopes (AFTE): Summarized and compared in [98], these feature sets are generally very useful for audio and music genre classification. MFCC feature set is very often used in audio fingerprinting applications as well. As claimed in [98], AFTE feature set is the most powerful for automatic genre classification. For a few particular audio classes, however, classification performance would be better if other feature sets (crowd noise: SLL and MFCC; classical music: SLL; speech: PA) are used.

- Chroma features: This set of features capture both melodic information and harmonic information. They are often used for the purpose of audio thumbnailing [5], audio fingerprinting or content-based audio identification (CBID) [46, 66] and audio matching [103].

Other than these three most popular feature groups, more and more novel feature sets have been introduced recently in the MIR research community to improve music retrieval performance. Here are a few examples:

The feature set of Daubechies Wavelet Coefficient Histograms (DWCH) of music signals was introduced in [85] to provide some extra information that the existing feature sets do not have. It is also used for the purpose of music genre classification and music recommendation.

Bass-line features were proposed in [145] and applied to automatic genre classification and music collection visualization. Bass lines are contained in many genres of music, and the type of rhythmic pulse used in bass lines varies widely in different types of music. It was claimed in [145] that the genre classification accuracy was improved by making use of these new features.

Trap-tandem features, which describe the timbre and rhythmic context of a note onset, were firstly adopted in [126] for music information retrieval. Several experiments were conducted in [126] and it was demonstrated that these new features were helpful to the application.

Music Retrieval Based on Symbolic Features

Using the features extracted from symbolic data, the symbolic-analytic approach treats music as sequences of notes and events making up a musical score. Symbolic data are mainly in the format of MIDI files. They can also be MusicXML, Humdrum or in other formats. MIDI, an abbreviation for Musical Instrument Digital Interface, is a criterion adopted by the electronic music industry for controlling devices, such as synthesizers and sound cards, that emit music. It is an encoding system representing, transferring and storing musical information. Instead of containing actual sound samples as audio encoding methods do, MIDI files store instructions that can be sent to synthesizers. The quality of sound produced when a MIDI file is played is therefore highly dependant on the synthesizer that the MIDI instructions are sent to. In effect, a MIDI recording gives us much the same information as we would find in a musical score. Therefore, MIDI, and other formats such as KERN, MusicXML or GUIDO, are often called symbolic formats.

Sometimes, music scores are also used in this analysis. Music score refers to a hand-written or printed form of musical notation, which normally uses a five-line staff to represent a piece of music work. The music scores are used in performing music pieces, for example, when a pianist plays a famous piano music. In the field of music data mining, some researchers focus on music score matching, score following and score alignment, to estimate the correspondence between audio data and symbolic score [32]. Some popular music score websites, *e.g.*, *music-scores.com*, provide music score downloading services.

As discussed in [96], conducting symbolic data analysis is to complement the analysis of audio data. The reasons of doing this can be summarized into the following [96]:

- It is hard to extract certain high-level features from audio data, such as precise note timing, voice and pitch. With symbolic data, we can achieve this.
- With audio data, due to the nature of the music recording, the processing speed and data amount, researchers normally just extract features from a very limited segment of the recording. While for symbolic analysis, it is possible to extract features of the entire file from the symbolic data.
- In certain cases, we have the music scores handy, but we do not have audio recordings available.

With symbolic features extracted from MIDI recordings, supervised learning techniques were selected to conduct music genre classification in [96] as those used for genre classification using audio data, because to keep a rule-based expert system is impractical, and to use unsupervised learning might generate clusters that do not make sense to human.

Music Retrieval Based on Lyrics Features

Lyrics are a set of words that make up a song in a textual format. In general, the meaning of the content underlying the lyrics might be explicit or implicit. Lyrics are very cultural-related, and most of them convey very specific meanings to music listeners. They can describe the artist's emotion, religious belief, or represent themes of times, beautiful natural scenery and so on. Well-written lyrics, such as poem-like lyrics, may significantly improve the attractiveness of the music work. The analysis of the correlation between lyrics and other music information may help us understand the intuition of the artists. Sometimes, lyrics might contain important hints, from which we can easily deduce the music genre, style and/or mood. On the World Wide Web, there are a couple of websites offering music lyrics searching services, e.g., *SmartLyrics* (`http://www.smartlyrics.com`) and *AZLyrics* (`http://www.azlyrics.com`). In music retrieval systems, lyrics should be considered as an important factor that affects the preferences of music listeners.

In [151], lyrics were used for keyword generation of songs, which can be applied to the application of automatic tagging. In [94], natural language processing techniques were applied to lyrics to perform interesting analysis like thematic categorization, and similarity searches of lyrics in music collections. In [68], non-negative matrix factorization (NMF) was employed to analyze lyrics and identify music topic clusters.

Lyrics are also utilized together with other music data such as audio data in music information retrieval research community. In [81], a semi-supervised learning approach was developed to analyze both lyrics and acoustic data to identify artist style. In [105], *Self-Organizing Maps* were used to combine audio features and song lyrics to organize audio collections and to display them via map-based interfaces. In our research study, we have also successfully made use of lyrics features. They have been used along with acoustic features to address the issue of clustering pop music into groups for the artists from diverse information sources. This effort will be briefly discussed in chapter 4.

Note that Bibliographic metadata [76] are not directly utilized by itself in the literature, but with its associated data such as artist information.

2.2.2 Social Context Data and Social-Context-based Music Retrieval

Social context data of music objects are often created by music consumers or experts manually.

Relational metadata [76] are generally created by music experts. As they are not derived directly from the music products, they are not unbiased, and can be heavily affected by cultural context. Music genres are the main annotations that music experts are trying to create for labeling and organizing music pieces. They are categories of music pieces that are closely related to music pieces, artists, culture and even the market. Different taxonomy of genres are adopted by different music stores to organize music collections. But for any set of genres, boundaries between the genres in the set are fuzzy [108]. Moods and styles are other terms used to describe music objects. Compared with genres, they are more descriptive and less abstractive, and in general one music piece can have many mood and style labels assigned. Such information of songs can be found at *AllMusic* (`http://www.allmusic.com`).

Mood and style descriptions of music pieces are valuable information for applications of music data organization and music recommendation, but to the best of our knowledge, not many MIR researches are making use of mood or style information. Therefore, we have conducted one study on music artist similarity measurement by utilizing the mood and style information extracted from *AllMusic*, which will be presented in chapter 3.

Associative Information [76] of music objects are often created by music consumers when they are listening to the music samples. As World Wide Web gets more and more popular, many music websites, such as `http://last.fm` and `http://music.strands.com`, are available for end users to generate such information on the websites. Below are the most available associative music data on the web, which have been utilized in a considerable amount of literatures:

- Music Reviews: Music reviews represent a rich resource for examining the ways that music fans describe their music preferences and possible impacts of those preferences. With the popularity of World Wide Web, an ever-increasing number of music fans are joining the music society and describing their attitudes towards music pieces. Online reviews can be surprisingly detailed, covering not only personal opinions of the reviewers but also important background and contextual information about the music piece and musicians [59].

- Music Social Tags: Music social tags are a collection of textual description that annotates different music items. The tags are typically used to facilitate searching for songs, exploring for new songs, locating similar music recordings, and finding other listeners with similar interests [74]. An illustrative example of well-known online music social tagging systems is *last.fm*, which provides plenty of music tags through public tagging activities.

- User Profiles and Playlists: User profile represents the user's preference to music information, *e.g.*, what kind of songs the user is interested in, and/or which artist the user favors. Playlist refers to the list of music pieces that the user created based on his/her preferences or the user has listened to. Traditionally, user profiles and playlists were stored in offline music applications, which could only be accessed by a single user. With the popularity of cyberspace and many music websites, more and more music listeners share their music preference online. As a result, the user profiles and playlists are now stored and managed in the online music databases, which are open to all the Internet users. Some popular online music applications, e.g., `http://www.playlist.com` and `http://music.strands.com`, provide services of creating user profiles and playlists, and sharing them on social networks.

Social-context-based music analysis generally use Collaborative Filtering (CF) approaches to work with social behavior data mined from popular websites. A great amount of efforts have been directed towards collecting textual correlations, and co-occurrences of music objects on public websites. For example, Schedl et al. [130] analyzed artist-based term co-occurrences based on web texts. Knees et al. [69] used semantic data mined from the results of web-searches for songs, albums and artists to generate a contextual description of the music based on large-scale social input. Whitman and Ellis [153] developed an unbiased music annotation system by leveraging web-mined reviews.

A very common problem in social-context-based music retrieval is derived from text mining. The retrieval can suffer from a lack of precision, and can be confused and not able to distinguish band/artist name terms from non-associated content, as mentioned in [130]. Music social tag information is also employed in our research study. In order to address the problem with text mining, we try to combine the social-context data with the music content data by using multi-label classification algorithm, which will be presented in chapter 3.

Playlists have been treated as a valuable information source of user access patterns in our research study. They have been utilized together with the acoustic features extracted from the music pieces for music recommendation purpose. This effort will be presented in chapter 5.

2.3 Music Retrieval Tasks and Common Techniques

In this section, we will review the following topics in the music retrieval literature and attempt to answer the question of what data mining algorithms or other related techniques are mostly used for the tasks of music data organization and annotation, music similarity search and recommendation, and retrieval result presentation. To be more exact, we will study the literature on the research issues of music genre classification, artist identification/classification, music similarity search and audio fingerprinting, user-centric music recommendation, and audio thumbnailing.

2.3.1 Music Genre Classification

Table 2 lists the common data mining and machine learning techniques employed in the music genre classification literature. The techniques are briefly introduced, and selected publications for each technique are also discussed.

Techniques	Publications
Bayesian Network	[36]
Decision Tree	[3]
Gaussian Mixture Model	[16]
Hidden Markov Model	[121]
K-Nearest Neighbor	[146], [110]
Linear Discriminant Analysis	[20]
Neural Networks	[75], [97]
Support Vector Machine	[77], [100]
Taxonomy	[82]
Multi-labeling Classification	[93]

Table 2: Various music genre classification techniques

- **Bayesian Network**. A Bayesian network is a graphical model that represents probabilistic relationships among variables of interest. In [36], Decoro et al. use a Bayesian framework to aggregate a hierarchy of multiple binary classifiers, which are support vector machines in the case of that publication, to generate music genre hierarchical taxonomies and improve classification accuracy.

- **Decision Tree**. A decision tree is a tree-like graph that represents decisions and their possible consequences. In [3], a musical piece is represented as a list of chords, and each musical genre as a series of musical pieces. Then a decision tree induction algorithm is adopted to find the patterns of chord sequences, that appear in many songs of one genre and do not appear in the other genres. Finally the discovered patterns are used to classify unknown musical pieces into genres.

- **Gaussian Mixture Model (GMM)**. GMM models the Probability Density Function (PDF) of observed variables using a multivariate Gaussian mixture density. Given a series of inputs, it refines the parameters for each Gaussian component and the mixture weights through iterative expectation-maximization (EM) algorithms. A 3-component Gaussian Mixture Model was used as classifier in [16] to perform genre classification task.

- **Hidden Markov Model (HMM)**. A hidden Markov model is a Markov chain for which the state is only partially observable. Hidden Markov models are especially known for their applications in temporal pattern recognition such as speed recognition. In [121], the acoustic segment model is employed to create a “timbre dictionary” , which is then used to train HMMs that represent the entire acoustic space for genre classification.

- **K-Nearest Neighbor (KNN)**. KNN is a method for classifying objects based on closest training examples in the feature space. The basic idea of KNN is to allow a small number of neighbors to influence the decision on a point. It is proven that the error of KNN is asymptotically at most twice as large as the Bayesian error rate. A number of standard statistical pattern recognition classifiers were used in [146] for comparison purposes, including KNN. [110] also uses KNN as its classifier.

- **Linear Discriminant Analysis (LDA)**. In the statistical pattern recognition literature discriminant analysis approaches are known to learn discriminative feature transformations very well. The basic idea of LDA is to find a linear combination of features which characterize or separate two or more classes of objects. LDA with Adaboost is used in [20] as the genre classifier in conjunction with other learning algorithms to improve their classification and generalization performances.

- **Neural Networks (NNs)**. Artificial Neural Networks (ANNs) are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data. The most widely used supervised ANN for pattern recognition is the Multi-Layer Perceptron (MLP). [75] uses MLP as the genre classifier, which is provided by WEKA, a machine learning tool. In [97], classification is performed using an ensemble of feedforward neural networks and k-nearest neighbour classifiers. It claims that the use of both techniques allows them to use Neural Networks to model the sophisticated relationships between features when required, while using KNN classifiers elsewhere to limit training times.

- **Support Vector Machines (SVMs)**. Support Vector Machines [149] aim at searching for a hyperplane or a set of hyperplanes that separate the positive data points and the negative data points with maximum margin. SVMs have demonstrated excellent performance in binary classification tasks. [77] and [100] use SVMs as their classifier for genre classification.

From the above description, we can see that most popular classification algorithms have been tried in the music genre classification research community. Advanced classification techniques have also been used in certain work, such as in [82], hierarchical classification with taxonomies was applied to music genre classification task, while SVMs were used to build the classifiers. The basic idea was to first classify audio excerpts into several genre groups that were a combination of several genres, then classify them into the desired genre within the genre groups. [93] tried to assign multi genre labels to music pieces. It decomposed the multi-label classification problem into multiple single-class classification problems by breaking it down in two dimensions. The classifier used was GMMs.

There have been as well a few comparative studies on different music genre classification algorithms. [85] compared the performances of SVM, KNN, GMM, and LDA applied to different acoustic features. [127] compared the performances of five modified methods based on three different classifiers (SVM, NN, and HMM). [128] provided a comprehensive survey on music genre classification topic. Expert systems were explained to be impractical, and different approaches in unsupervised learning and supervised learning systems were discussed.

Specifically in our research effort, we have proposed a multi-label classification approach, called *Hypergraph integrated Support Vector Machine (HiSVM)*. It can integrate several types of music information including music audio features, music style correlations, and social tag information and correlations.

2.3.2 Artist Identification/Classification

Automated artist identification is important for many MIR applications including music indexing and retrieval, copyright management and music recommendation systems. The development of artist identification enables the effective management of music databases based on "artist similarity". Automatic artist classification refers to classifying musicians as the predefined artist label given a music document. Most often, artist identification/classification is performed based on acoustic features of the singer voice. Sometimes, social context data such as web data can be used for this purpose [69].

Table 3 lists the common data mining and machine learning techniques employed in the automatic artist identification/classification literature.

Techniques	Publications
Gaussian Mixture Model	[50], [67], [136], [144], [161]
K-Nearest Neighbor	[88]
Neural Networks	[7]
Support Vector Machine	[69]

Table 3: Various artist identification/classification techniques

In this dissertation, we investigated a new approach to quantify the music artist similarity by employing the artist style and mood information extracted from All Music Guide. A hierarchical co-clustering algorithm was adopted for this study.

2.3.3 Music Similarity Search

In the field of music data mining, similarity search refers to searching for music sound files/samples similar to a given music sound file/sample. In principle, searching can be carried out on any dimension. For instance, the user could provide an example of the timbre, or of the sound, that he is looking for, or describe the particular structure of a song, and then the music search system will be search similar music works based the information given by the user.

The similarity search processes can be divided into feature extraction and query processing [80]. Feature extraction is the procedure to extract the content features described in section 2.2.1. Some feature extraction procedures or instructions will be explained in chapter 3. After feature extraction, music works can be represented based on the extracted features. In the step of query processing, the main task is to employ a proper similarity measure to calculate the similarity between the given music work and the candidate music works. A variety of existing similarity measures and distance functions have previously been examined in this context, spanning from simple Euclidean and Mahalanobis distances in feature space to information theoretic measures like the Earth Mover Distance and Kullback-Leibler [8]. Regardless of the final measure, a major trend in the music retrieval community has been to use a density model of the features (often timbre space defined by MFCC's [118]).

Numerous data mining and machine learning approaches have been applied to the problem of music similarity search task. Rauber et al. studied a hierarchical approach in retrieving similar music sounds [120]. Schnitzer et al. re-scaled the divergence and used a modified FastMap implementation to accelerate nearest-neighbor queries [132]. Slaney et al. learned embeddings so that the pairwise Euclidean distance between two songs reflected semantic dissimilarity [137]. Deliège et al. performed the feature extraction in a two-step process that allowed distributed computations while respecting copyright laws [38]. In [80], Li et al. defined the distance between two sound files to be the Euclidean distance of the normalized representations of acoustic features. Pampalk et al. presented an approach to improve audio-based music similarity and genre classification [110]. Berenzweig et al. examined both acoustic and subjective approaches for calculating similarity between artists, comparing their performance on a common database of 400 popular artists [8]. Aucouturier et al. introduced a timbral similarity measures for comparing music titles based on a Gaussian model of cepstrum coefficients [4].

As a special type of music similarity search, audio fingerprinting is best known for its ability to associate an unlabeled music piece with its singer and track title. Compared to other music similarity search applications, audio fingerprinting applications or audio identification applications try to search an exact match of a given audio input in a large music database. Similar to the music similarity search, there are two fundamental processes in audio fingerprinting: the fingerprint extraction and the matching algorithm [18].

Fingerprint extraction is the process of extracting content features from the audio data, or cryptographic audio hashing of the audio data to represent the music piece. Applications using feature extractions for audio fingerprinting work very similarly to other similarity search applications. MFCC features [6], and chroma features [46] are the normal selections for this purpose. Systems based on audio hashing are generally designed case by case in this process. [53] extracted 32-bit hash value for every frame and represented the music pieces as binary vector sequences. The fingerprint used in [112] was a sequence of vectors representing band information.

There were numerous matching algorithms used in this context. The identification system in [6] was built on Hidden Markov Models (HMM). [53] designed an efficient bit matching algorithm to search the audio in the music database. [19] adopted an approach used in computational biology for the comparison of DNA to accelerate the search speed of fingerprints.

2.3.4 User-Centric Music Recommendation

Music recommendation aims to provide a music listener a list of music pieces that he/she is likely to enjoy. User-centric music recommendation should focus on the user to whom the system is intended to deliver the retrieval results, and therefore should be based on a good understanding of the user preferences as well as the music pieces in the collection.

Various music recommendation approaches have been developed, and user demographic information, music contents, user listening history, and the discography (e.g., Last.fm, Goombah, and Pandora) have been used for music recommendations [17, 24, 89, 106, 109, 113, 116, 147]. These approaches can be generally divided into two groups: collaborative-filtering methods and content-based methods.

Collaborative-filtering methods recommend songs by identifying similar users or items based on ratings of items given by users [14, 28, 57]. If the rating of an item by a user is unavailable, collaborative-filtering methods estimate it by computing a weighted average of known ratings of the items from similar users. Thus, for collaborative-filtering methods to be effective, large amount of user-rating data are required, which is a major limitation [125, 129]. *Content-based methods* provide recommendations based on the meta-data such as genre, styles, artists, and lyrics [113, 119], and/or the acoustic features extracted from audio samples [60,70,78,80]. Since acoustic contents are susceptible to feature extraction, music recommendation is considered different from movie recommendation, in which meta-data is generally the only available information [99]. In music recommendation, the reflective and consistent acoustic features can represent song-specific characteristics such as genre, timbre, pitch, and rhythm. Comparing with the acoustic features, a large portion of meta-data are the descriptions of contents given by musicians. Music meta-data are thus very time-consuming to obtain and not capable of providing adequate information for describing listeners' preferences [78].

Recently probabilistic models and hybrid algorithms [65, 117, 159] have been proposed to overcome the aforementioned limitations by combining contents and user ratings. Yoshii et al. [159] attempted to integrate both rating and content data. They utilized Bayesian network to statistically estimate the probabilistic relations over users, ratings and contents. Popescul et al. [117] proposed a probabilistic model similar to the one suggested by Yoshii et al. to take advantage of both collaborative-filtering and content-based recommendations. Jung et al. [65] designed a hybrid method that combines collaborative-filtering and content-based methods to improve recommendation performance. However, these models and methods significantly degraded when they were short of corresponding user access data as illustrated in our experiments that will be presented in chapter 5.

In this dissertation, user-centric music recommendation is one of the major research issue. We propose a music recommendation approach by incorporating collaborative-filtering and acoustic contents of music. This approach employs a novel dynamic music similarity measurement strategy, which significantly improves the similarity measurement in terms of accuracy and efficiency. This measurement strategy utilizes the user access patterns from large numbers of users and represents music similarity with an undirected graph. Recommendation is calculated using the graph Laplacian and label propagation defined over the graph. More details can be found in chapter 5.

2.3.5 Music Audio Thumbnailing

Audio Thumbnailing, also called Music thumbnailing or music summarization, aims at finding the most representative part of a song, which can be used for music browsing, music searching and music recommendation. In this context, if the music retrieval results tend to present a long list, summarized music pieces would be very helpful for the end users to digest the information.

[72] presented a music summarization system developed on MIDI format, which utilized the repetition nature of MIDI compositions to automatically recognize the main melody theme segment and generate music summary. [58] also proposed two approaches dealing with symbolic data for music thumbnailing purpose.

However, most such studies have been worked on music audio signals. Logan et al. [90] tried to use a clustering technique or Hidden Markov Models to find key phrases of songs.

MFCC features were selected to parameterize each music song. This summarization method was suitable for certain genres of music such as rock or folk music, but it was less applicable to classical music. MFCCs were also used as features in Cooper and Foote's works [30, 31]. They used a two-dimensional similarity matrix to represent music structure and generate music summary. But this approach would not always yield intuitive music pieces. In [21] and [22], Chai and Vercoe presented a structural analysis method with five steps, which were: 1) feature extraction, 2) pattern matching, 3) repetition detection, 4) segment merging, and 5) structure labeling. Peeters et al. [114] proposed a multi-pass approach to generate music summaries. [157] proposed effective algorithms to automatically classify and summarize music content. Support vector machines were applied to classify music into pure music and vocal music by learning from training data.

In the prototype system we developed in this study, the approach proposed in [30] was adopted to create audio thumbnails, which were continuous excerpts of the whole music pieces.

CHAPTER 3

MUSIC DATA ORGANIZATION AND ANNOTATION

3.1 Introduction

Huron [61] points out that, because the preeminent functions of music are social and psychological, the most useful characterization of music would be based on a variety of social context information [76] including genre, style, mood, and similarity. Therefore, to enable music queries, it is imperative that each piece of music be annotated by appropriate labels, not only by artists, album titles, track titles, and genres, which in many cases are readily available at GraceNote, an access-free on-line database, but also by more pertinent information, such as style and mood labels, which are available at online music stores and AllMusic, a registered-user-only on-line database. Music annotation considers the problem of automatically assigning the latter type of labels so as to eliminate the need of accessing those limited-access databases. Recently, the user-assigned tags have turned into an essential component in music information retrieval research and the problem of automatic music annotation using user-assigned tags has also attracted a lot of research attention.

We study two important problems for music data organization and annotation. The first problem is correlating styles and mood labels. An important characteristic of the style and mood labels is that most labels are having close semantic relationships. One challenge is whether we can automatically characterize the semantic relations among the labels using a hierarchical structure. In this dissertation, we develop a hierarchical divisive co-clustering algorithm for exploring the relationships among the style/mood models. The discovered relationship can be used to compute the similarity between music artists. The second problem is multi-label mood/style classification. Traditional music mood/style classification approaches usually assume that each piece of music has a unique mood/style and they make use of the music content (audio features) to construct a classifier for classifying each piece into its unique mood/style. However, in reality, a piece of music may match more than one, even several different moods/styles. In addition, how to incorporate the tag information into the classification process is also a challenge. In this dissertation, we develop a novel multi-label music mood/style classification approach with hypergraph regularization to address this problem.

The rest of this chapter is organized as follows: Section 3.2 introduces audio feature extraction that is useful for subsequent analysis, Section 3.3 presents the study of quantifying the artist similarity via a hierarchical divisive co-clustering algorithm of the style/mood models, and Section 3.4 describes the developed multi-label mood/style classification algorithm with hypergraph regularization.

3.2 Audio Feature Extraction

Before applying data mining approaches into music information retrieval tasks, an important step is the determination of the features extracted from music data. All the machine learning methods discussed in this chapter and the following chapters are making use of the acoustic content features extracted from music audio signals to certain extent. There has been a considerable amount of work in extracting descriptive features from music signals for music genre classification and artist identification. In our study, we use timbral features along with wavelet coefficient histograms. The feature set consists of the following three parts and total 80 features, which can efficiently reflect the moods and styles of the corresponding artists [49, 85, 91, 146].

- Mel-Frequency Cepstral Coefficients (MFCC): MFCC is a feature set that is highly popular in speech processing. It is designed to capture short-term spectral-based features. The features are computed as follows: First, for each frame, the logarithm of the amplitude spectrum based on short-term Fourier transform is calculated, where the frequencies are divided into thirteen bins using the Mel-frequency scaling. Next, this vector is decorrelated using discrete cosine transform. This is the MFCC vector. In this work, we use the first five bins, and compute the mean and variance of each over the frames.

- Short-Term Fourier Transform Features (STFT): This is a set of features related to timbral textures and is not captured using MFCC. It consists of the following five types: Spectral Centroid, Spectral Rolloff, Spectral Flux, Zero Crossings, and Low Energy. More detailed descriptions of STFT can be found in [146].

- Daubechies Wavelet Coefficient Histograms (DWCH): Daubechies wavelet filters are a set of filters that are popular in image retrieval. The Daubechies Wavelet Coefficient Histograms, proposed in [85], are features extracted in the following manner: First, the Daubechies-8 (db_8) filter with seven levels of decomposition (or seven subbands) is applied to 30 seconds of monaural audio signals. Then, the histogram of the wavelet coefficients is computed at each subband. Following that, the first three moments of a histogram, i.e., the average, the variance, and the skewness, are calculated from each subband. In addition, the subband energy, defined as the mean of the absolute value of the coefficients, is computed from each subband. More details of DWCH can be found in [84, 85].

3.3 Quantify Music Artist Similarity Based on Style and Mood

3.3.1 Introduction

Music artist similarity has been an active research topic in music information retrieval area for a long time since it is especially useful for music recommendation and data organization [47, 84]. Many characteristics can be brought into consideration for defining similarity, e.g., sound, lyrics, genre, style, and mood. Methods for calculating artistic similarity include recent proposals that are based on the similarity information provided by the *All Music Guide* website (`http://www.allmusic.com`) as well as those based on the user access history (e.g., see [47]). Although there have been considerable efforts into developing effective and efficient methods for calculating artist similarity, several challenges still exist. First, artist similarity varies significantly when considering different aspects of artists such as genre, mood, style, culture, and acoustics. Second, the user access history data are often very sparse and hard to acquire if we are to calculate the artist similarity based on the user access history.

Third, even if we can obtain the categorical descriptions of two artists using *All Music Guide*, correlating and comparing the descriptions is not trivial since there are semantic similarities among different descriptions. For example, given two mood terms *witty* and *thoughtful*, we cannot simply quantify their similarity as 0 just because they are different words or as 1 because they are synonyms.

In this section, we propose a new framework for quantifying artist similarity. In this framework, we focus on two important aspects of music: style and mood [147]. The style and mood descriptions of famous artists are publicly available on *All Music Guide* website. We collect the information of the artists and their style and mood descriptions. The *All Music Guide* style terms are nouns and adjectives while its mood terms are adjectives only. These terms carry significant linguistic meanings given some context, but the use of the terms at the *All Music Guide* web site is little contextual. In this research work, we study how these terms are collectively used in describing artists.

3.3.2 Hierarchical Co-clustering of style and mood labels

An important characteristic of the style and mood labels is that most labels are having close semantic relationships. The first type of the relationships is that some labels are synonyms, e.g., “witty” and “thoughtful” are synonyms, and “happy” and “cheerful” are synonyms as well. The second type of the relationships is that some labels are more general while some others are more specific, e.g., “Soft Metal” is a more specific style than “Metal”, “Extremely Provocative” is a more specific tag than “Provocative”, and “Agony” is a more specific mood label than “Sadness”. One challenge is whether we can automatically characterize such two types of semantic relations among the labels using a hierarchical structure.

To capture the semantic similarity among different style and mood descriptions, we generate a style taxonomy and a mood taxonomy using a hierarchical co-clustering algorithm. Then we quantify the semantic similarities among the style/mood terms based on the taxonomy structure and the positions of these terms in the taxonomies. Finally we calculate the artist similarities according to all the style/mood terms used to describe the music artists.

Similarity Taxonomy Generation

The style and mood labels of 2431 artists are collected for those artists having both labels appearing on the *All Music Guide* website. Altogether 601 style terms (nouns like *Electric Chicago Blues*, *Greek Folk*, and *Chinese Pop*, as well as adjectives like *Joyous*, *Energetic*, and *New Romantic*), and 254 mood terms (such adjectives as *happy*, *sad*, and *delicate*) are used to describe these artists. Table 4 lists an example of the mood and style descriptions of three randomly picked artists: ABBA, The Beatles, and Elvis Presley. The mood terms and style terms are subjective. However, we consider the labels of moods and styles from *All Music Guide* as representing collective opinions of many music experts/critics thereby representing the subjective opinions of a large proportion of music listeners.

Artist	Mood Description	Style Description
ABBA	Light, Delicate, Rousing, Sentimental, Joyous, Fun, Sweet, Sparkling, Sugary, Cheerful, Happy, Playful, Naive, Plaintive, Gentle, Gleeful, Giddy, Stylish, Romantic, Energetic, Exuberant, Ambitious, Complex, Exciting, Fun, Bright, Lively, Witty, Carefree, Wistful	Euro-Pop, Pop/Rock, Swedish Pop/Rock, Pop, British Invasion, Psychedelic
The Beatles	Wistful, Searching, Sweet, Warm, Yearning, Whimsical, Amiable/Good-Natured, Poignant, Lush, Laid-Back/Mellow, Literate	Merseybeat, Pop/Rock, British Psychedelia, Folk-Rock, Rock & Roll
Elvis Presley	Rock & Roll, Rockabilly, Pop, Pop/Rock	Carefree, Dramatic, Exciting, Confident, Exuberant, Energetic, Summery, Joyous, Rambunctious, Bright, Light, Romantic, Cheerful, Freewheeling, Raucous, Sweet, Playful, Fun, Warm, Swaggering, Lively

Table 4: An example of artist mood and style descriptions

Algorithm Description

In order to organize the style terms and mood terms into the corresponding taxonomies, we need to apply clustering algorithms to them. Clustering is the problem of partitioning a finite set of points in a multi-dimensional space into classes (called clusters) so that (i) the points belonging to the same class are *similar* and (ii) the points belonging to different classes are *dissimilar* [54, 62].

However, most clustering algorithms aim at clustering *homogeneous data*, i.e, the data points of a single type [10]. In our application, the data set to be analyzed involves more than one type, e.g. *styles* and *artists*. Furthermore, there are close relationships between these types of data. It is difficult for the traditional clustering algorithms to utilize those relationship information efficiently.

Co-clustering algorithms are designed to cluster different types of data simultaneously by making use of the dual relationship information such as mood–artist matrix. For instance, *Dhillon* [40] and Zha et al [160] proposed bipartite spectral graph partitioning approaches to co-cluster words and documents, *Cho. et al* [25] proposed to co-cluster the experimental conditions and genes for microarray data by minimizing the *Sum-Squared Residue*, *Long et al.* [92] proposed a general principled model, called *Relation Summary Network*, to co-cluster the heterogeneous data on a *k-partite graph*.

On the other hand, hierarchical clustering is the problem of organizing data in a tree-like structure in which the input set of data points is recursively divided into smaller subgroups, usually until the subgroups become individual data points [142]. While both hierarchical clustering and co-clustering have their own advantages, few algorithms exist that execute both simultaneously [11]. In our work, to further utilize the cluster information obtained from the co-clustering algorithms and generate the taxonomies, we utilize a hierarchial co-clustering algorithm [158].

In our work, the artist style description is represented as a 2431×601 artist–style matrix, S, and the artist mood description as a 2431×254 artist–mood matrix, M. In the following, we will describe our algorithm for the artist–style matrix S. The algorithm is the same for the artist–mood matrix M. The core idea behind the procedure is to combine Singular Value Decomposition (SVD) and K-means using a top-down iterative process [158]. The procedure is described as follows:

1. Given an $m \times n$ artist–style matrix, S, perform SVD on S to obtain: $S = U \times \Lambda \times V^T$.

2. Let $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_m$ be the largest m singular values. Then the number of clusters is k where:
$$k = argmax_{(m \geqslant i > 1)}(\lambda_{i-1} - \lambda_i)/\lambda_{i-1}.$$

3. Find k singular vectors of S: $u_1, u_2, \ldots, u_k$ and $v_1, v_2, \ldots, v_k$, and then form a matrix Z by:
$$Z = \begin{bmatrix} D_1^{-1/2}[u_1, ..., u_k] \\ D_2^{-1/2}[v_1, ..., v_k] \end{bmatrix}.$$

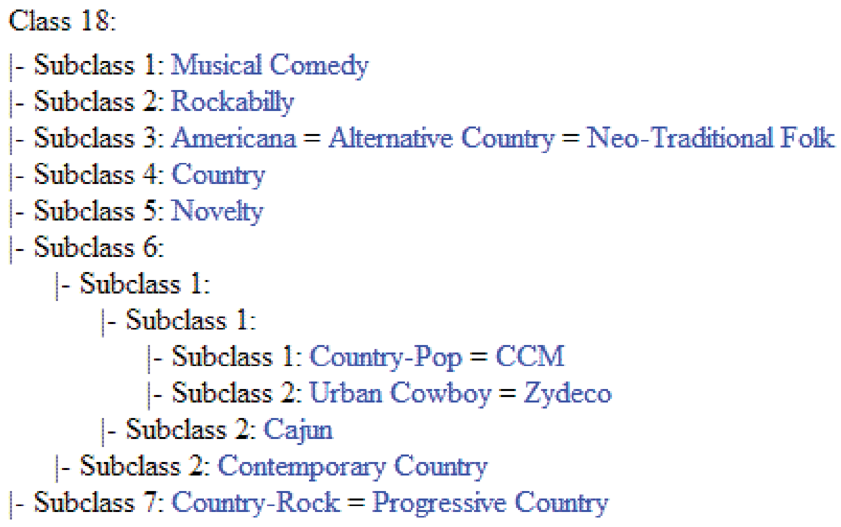
4. Apply K-means clustering algorithm to cluster Z into k clusters.

5. For each cluster, check the number of artists in it. If the number is higher than a given threshold (in our experiment, we set the threshold = 3), construct a new artist–style matrix formed by the artists and styles in that cluster, and continue to the first step.

According to this algorithm, 601 style terms are clustered into 20 clusters, and 254 mood terms are clustered into 68 clusters. They are further recursively clustered into many subclasses until the algorithm converges. We organized the generated taxonomies and present them in two trees, which can be viewed at `http://www.newwisdom.net/MIR/styletree.jsp` and `http://www.newwisdom.net/MIR/moodtree.jsp`.

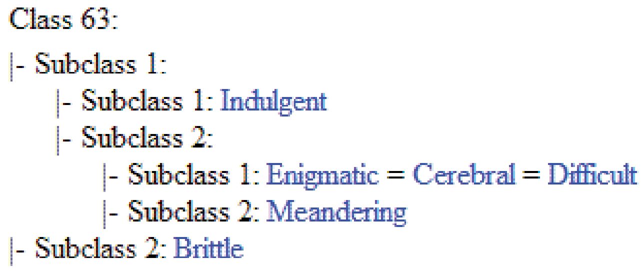
Figure 1 is an example of a style cluster obtained from the style similarity tree. By checking the positions of the terms in the taxonomy of this cluster, we can easily observe that *Country-Rock* and *Progressive Country* are the most similar (similarity value between them equals to 1 in our system) in the semantic meanings of styles, and the similarity between *Country-Pop* and *Urban Cowboy* is greater than the similarity between *Country-Pop* and *Cajun* as well as the similarity between *Urban Cowboy* and *Cajun*. Figure 2 is an example of a mood cluster obtained from the mood similarity tree that is generated following the same construction rule.

Figure 3 shows the distribution of the sizes of all the 20 style clusters and Figure 4 shows the



```
Class 18:
|- Subclass 1: Musical Comedy
|- Subclass 2: Rockabilly
|- Subclass 3: Americana = Alternative Country = Neo-Traditional Folk
|- Subclass 4: Country
|- Subclass 5: Novelty
|- Subclass 6:
    |- Subclass 1:
        |- Subclass 1:
            |- Subclass 1: Country-Pop = CCM
            |- Subclass 2: Urban Cowboy = Zydeco
        |- Subclass 2: Cajun
    |- Subclass 2: Contemporary Country
|- Subclass 7: Country-Rock = Progressive Country
```

Figure 1: An example of a style cluster from the style similarity tree (= means the most similar)



```
Class 63:
|- Subclass 1:
    |- Subclass 1: Indulgent
    |- Subclass 2:
        |- Subclass 1: Enigmatic = Cerebral = Difficult
        |- Subclass 2: Meandering
|- Subclass 2: Brittle
```

Figure 2: An example of a mood cluster from the mood similarity tree (= means the most similar)

distribution of the sizes of all the 68 mood clusters. From these two figures, we observe that both style terms and mood terms are distributed into each classes in a quite balanced manner.

Based on this co-clustering algorithm, we can also obtain the style-based artist similarity structure and mood-based artist similarity structure directly, which can be viewed at `http://www.newwisdom.net/MIR/artisttrees.jsp` and `http://www.newwisdom.net/MIR/artisttreem.jsp`. They have the similar well-balanced cluster member distributions.

However, the similarity between two artists has not been quantified up to now. Furthermore, if we have new artists with style and/or mood descriptions, it is very hard for us to integrated them into the tree structures. Therefore, we need to go steps further to study how to quantify the term similarity and artist similarity based on the generated taxonomies.

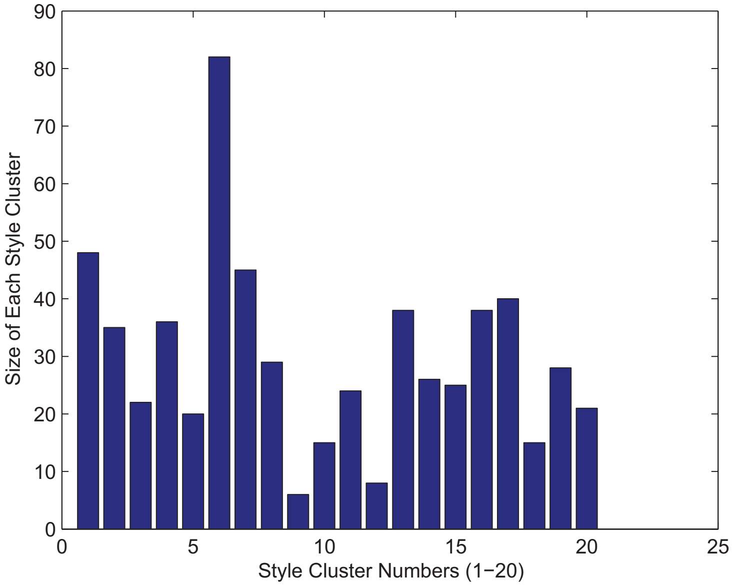


Figure 3: The distribution of sizes of style clusters

3.3.3 *Similarity Quantification*

To calculate artist similarity, we need to quantify the semantic similarity between all pairs of style/mood terms first. In order to do this, we investigate the methods proposed by Resnik [122], Jiang and Conrath [63], Lin [87], and Schlicker et al. [131]. The approaches for calculating the similarity proposed by them are briefly described as follows:

Resnik:

$$sim_R(s_1,s_2) = \max_{s \in S(s_1,s_2)} \{-\log(p(s))\}. \tag{1}$$

Jiang-Conrath:

$$\begin{aligned} & dist_{JC}(s_1,s_2) \\ = & \max_{s \in S(s_1,s_2)} \{2\log(p(s)) - \log(p(s_1)) - \log(p(s_2))\}. \end{aligned} \tag{2}$$

Lin:

$$sim_L(s_1,s_2) = \max_{s \in S(s_1,s_2)} \{\frac{2 \times \log(p(s))}{\log(p(s_1)) + \log(p(s_2))}\}. \tag{3}$$

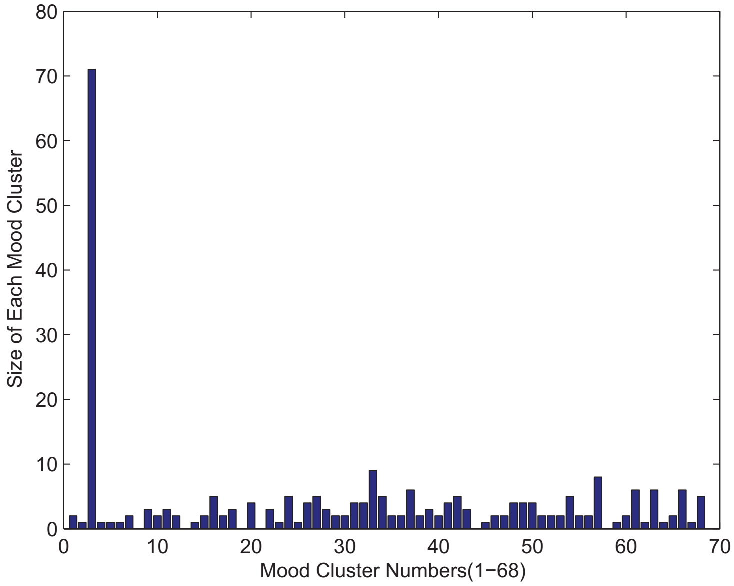


Figure 4: The distribution of sizes of mood clusters

Schlicker:

$$
\begin{aligned}
& sim_S(s_1, s_2) && (4)\\
= & \max_{s \in S(s_1,s_2)} \{ \frac{2 \times \log(p(s))}{\log(p(s_1)) + \log(p(s_2))} (1 - \log(p(s))) \}.
\end{aligned}
$$

Here $p(s) = freq(s)/N$ and $freq(s)$ is the number of artists that utilize the given style/mood term s to describe them, N is total number of artists, and $S(s_1, s_2)$ is the set of common subsumers of style/mood terms s_1 and s_2. The basic idea of these approaches is to capture the specificity of each style/mood term and to calculate the similarity between style/mood terms that reflects their positions in the taxonomy generated in Section 3.3.2.

Once we obtain the pairwise semantic similarity of style/mood terms, we can calculate the artist similarity based on style/mood. For example, if artist A_1 is described by a group of styles $s_1, s_2, \ldots, s_i$, and artist A_2 is described by another group of styles $s'_1, s'_2, \ldots, s'_j$, we define the style-based similarity between A_1 and A_2 as:

$$sim(A_1, A_2) = \frac{\sum_{x \in [1,i]} (max_{y \in [1,j]} sim(s_x, s'_y))}{j}. \quad (5)$$

Here $sim(s_x, s'_y)$ is the similarity between style s_x and style s'_y. Mood-based artist similarity can be obtained using the same approach.

In some applications, people may see the differences among these four different approaches due to the different scales of their results and the different ways they are associating with the terms in the taxonomies. In our system, however, we compared their results and do not observe any significant differences among them after normalizing them into the same scale (0∼1). To further illustrate this, let us check the data distribution of the artist similarity values generated using these four different approaches.

The distribution of artist similarity values based on style similarity calculated using the four different approaches is presented in Figure 5, and the distribution of artist similarity values based on mood similarity calculated using the four different approaches is presented in Figure 6.

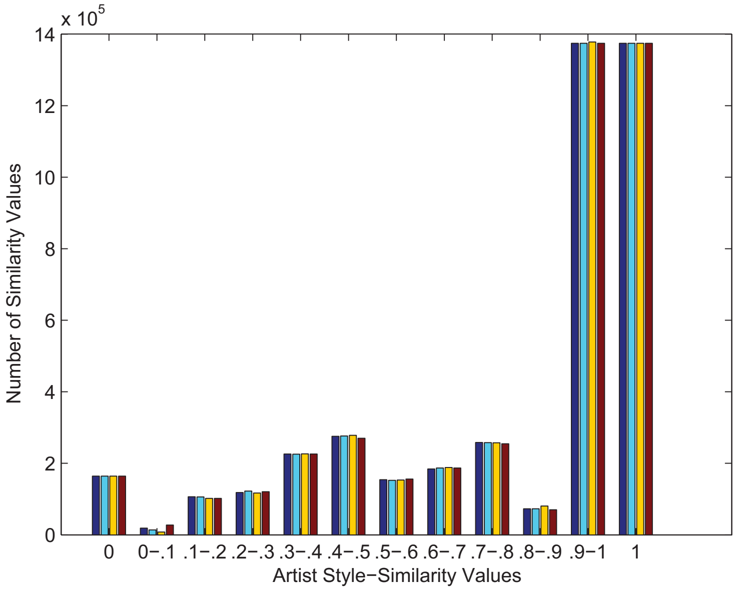


Figure 5: The distribution of artist similarity values based on style similarity calculated using the four different approaches

From these two figures, we observe that there are almost no difference among the distributions of the artist similarity values using 4 different approaches described above. Hence we use the average of all the 4 normalized quantified similarity values as the final artist similarity. We also observe that the style-based artist similarity values are a little more diverse

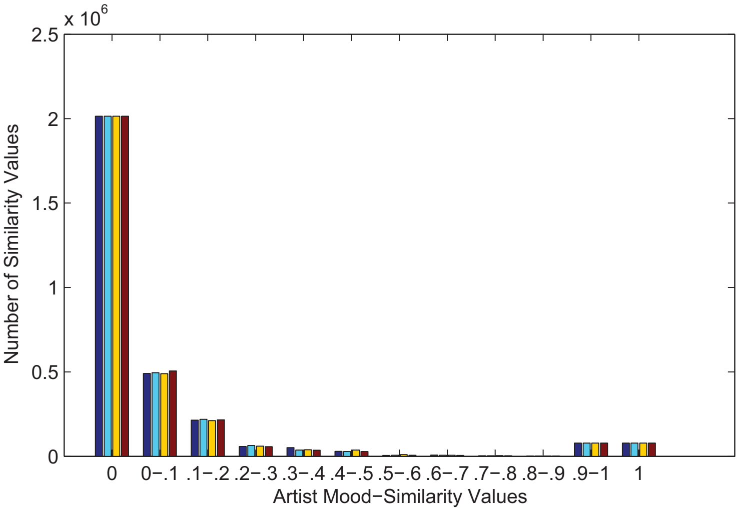


Figure 6: The distribution of artist similarity values based on mood similarity calculated using the four different approaches

than the mood-based artist similarity values, therefore we use a heuristic proportion value to calculate the final combined artist similarity value:

$$c = 0.4 \times m + 0.6 \times s, \tag{6}$$

where c is the combined artist similarity, and m is mood-based similarity while s is style-based similarity. In our system, "0" stands for the most different and "1" the most similar.

3.3.4 Evaluation

For the evaluation purpose, we are interested in how these *professionally assigned* mood and style terms are grouped together in describing artists. We believe that neither acoustic similarity nor mood/style labels provide sufficient information to enable accurate similarity calculation. We are rather interested in how related the label-based similarity and the acoustics-based similarity are to each other. To explore more on this question, it would be ideal if we had acoustics data for all the 2431 artists in the study, but the time and cost required for collecting the data would be prohibitive. So for this experimental study, we consider a limited number of artists to demonstrate the effectiveness of our framework. The case study we conducted is based on six famous artists (bands): the Beatles, the Carpenters, Celine Dion, Elvis Presley, Madonna, and Michael Jackson. The quantified artist similarities among them are listed in the second, third, and fourth columns of Table 5.

To compare with the artist similarity based on the mood and style labels, we use the distances of the acoustic features extracted from the songs of these artists (bands). For each artist (band), we randomly pick 5 songs and conduct the following procedure. Firstly, we exact the acoustic features of each song using the approach explained earlier in this chapter. Then we calculate the pairwise Euclidean distances between the feature points that represent the songs of different artists (bands). Finally we calculate the average of all the pairwise distances as the content-based distance of the two artists. The results are listed in the last column of Table 5.

Name Pair	Mood-based Similarity	Style-based Similarity	Combined Similarity	Average Distance
Elvis Presley : Michael Jackson	0.33	1	0.732	4.807
The Carpenters : Celine Dion	0.15	1	0.66	4.836
Michael Jackson : Madonna	0	1	0.6	6.840
The Carpenters : Michael Jackson	0	1	0.6	7.921
Celine Dion : Michael Jackson	0	1	0.6	7.991
Elvis Presley : Madonna	0	1	0.6	8.555
The Beatles : Michael Jackson	0	0.875	0.525	9.455
Celine Dion : Madonna	0	0.75	0.45	8.229
The Carpenters : Madonna	0.143	0.5	0.357	8.344
Celine Dion : Elvis Presley	0	0.75	0.45	8.655
The Carpenters : Elvis Presley	0.048	0.5	0.319	8.756
The Beatles : Madonna	0	0.5	0.3	9.688
The Beatles : Elvis Presley	0	0.5	0.3	9.324
The Beatles : Celine Dion	0	0.278	0.167	10.887
The Carpenters : The Beatles	0	0.25	0.15	11.134

Table 5: Quantified similarity values and average distances

In this case study, we also evaluate the sensitivity of the heuristic values used to calculate the combined similarity from the style-based similarity and mood-based similarity. They are based on the same artists (bands). Combinations based on different heuristic values are illustrated in Figure 7. In this figure, the values of content-based distances are decreased to 10% of their original values to fit into the scale. *0.2 + 0.8 combination* stands for $c = 0.2 \times m + 0.8 \times s$, where c is the combined artist similarity, and m is mood-based similarity while s is style-based similarity. All other types of combined similarities are calculated following the same rule.

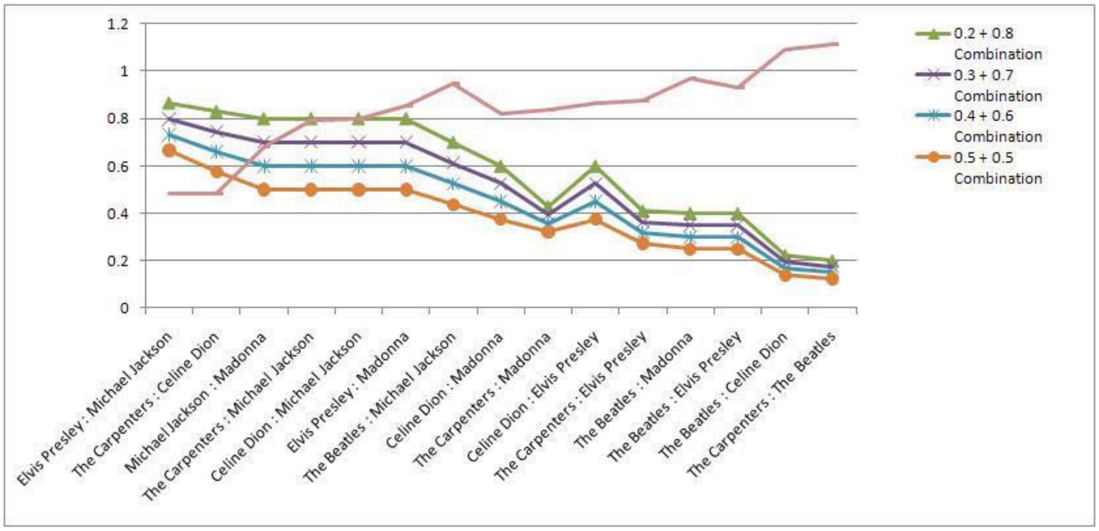


Figure 7: Comparison of combined similarities based on different heuristic values

Result Analysis

From the results, we observe that our quantified artist similarities match very closely the artist similarities based on the acoustic features extracted from the music recordings of the corresponding artists. By checking the last two columns of Table 5, we can easily observe that the data variation trends from the top to the bottom, i.e, while the average distance increases one by one, the combined similarity decreases almost constantly. In other words, the acoustic feature points of songs from the artists with higher similarity values (e.g., The Carpenters versus Celine Dion) are closer than those of songs from the artists with lower similarity values (e.g., The Beatles versus Celine Dion, and The Beatles versus The Carpenters), while the acoustic feature points of songs from the artists with lower similarity values (e.g., Elvis Presley and The Beatles) are farther than those of songs from the artists with higher similarity values (e.g., Elvis Presley and Michael Jackson). This demonstrates that our quantified artist similarity based on style and mood descriptions is consistent with the content-based artist similarity.

By checking Figure 7, we can observe that the calculation of combined artist similarity is not sensitive to the heuristic values used. All four combinations are having the same decreasing pattern when the content-based distance increases. This result indicates that it is possible to choose any of these four combinations to calculate the combined artist similarity given the style-based similarity and mood-based similarity. However, in order to let the combined similarity be able to reflect most of the information in the two components and the more diversified nature of style-based similarity, we use the *0.4 + 0.6 combination* in our system.

3.3.5 Conclusion

Music artist similarity has been an active research topic in music information retrieval for a long time since it is especially useful for music recommendation and organization. But artist similarity varies from different aspects considered, and is hard to quantify although considerable efforts have been put into this venue. In this investigated approach, we focus on two very important aspects of musical artists: style and mood. we extract authoritative information from All Music Guide, generate style and mood similarity taxonomies to represent the semantic relations among the style and mood terms, and quantify the artist similarities based on the semantic similarities of the style and mood terms. We also conduct a case study based on acoustic content analysis, which validates this quantification approach and shows the effectiveness of our proposed framework.

3.4 Tag Integrated Multi-label Music Style Classification with Hypergraph

Music genre and style classification has been a hot topic in Music Information Retrieval research area, and a significant amount of efforts have been put in this venue [85, 164]. Many approaches are highly successful, however there are two major limitations: 1) Most of them are single-label methods in that they can assign only one genre or style label to the music object, but many music pieces may map to more than one genre or style; 2) They mostly only make use of the music content information, which actually ignores the essential social context information of the music object.

In our work, we propose a SVM-like multi-label music style classification approach, called *Hypergraph integrated Support Vector Machine (HiSVM)*. The algorithm employs a hypergraph Laplacian regularizer and the problem can be efficiently solved by the dual coordinate descent method. The proposed approach can effectively perform multi-label music style classification by integrating three type of information: 1) audio features; 2) music style correlations; and 3) social tag information and correlations.

3.4.1 Method description

We build two hypergraphs in this work. A *hypergraph* is a generalization of a graph, where an *edge* can connect any positive number of *vertices* [9]. Formally, a hypergraph $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of *vertices* and $\mathcal{E} \in 2^{\mathcal{V}} - \Phi$ is a set of *edges*. An *edge-weighted hypergraph* is a hypergraph in which each edge is assigned a weight. Let us use *w(e)* to denote the weight given to an edge e. The *degree* of an edge e, denoted as $\delta(e)$, is the number of vertices connected to e. Thus for a standard graph ("2-graphs") the value of δ is 2 for all edges. The degree of a vertex v is $d(v) = \Sigma_{v \in e, e \in (e)} w(e)$.

The two hypergraghs we constructed in our music style classification are: the style hypergraph $\mathcal{G}_s$ and the tag hypergraph $\mathcal{G}_t$. The vertices of $\mathcal{G}_s$ and $\mathcal{G}_t$ are simply the data points. The hyperedges of $\mathcal{G}_s$ correspond to the style labels, i.e., each hyperedge in $\mathcal{G}_s$ contains all the data points that belong to a specific style category. Similarly, each hyperedge of $\mathcal{G}_t$ contains all the data points that own the corresponding tag.

Figure 8 shows an intuitive example on the music style and tag hypergraphs. In the figure, the nodes on the hypergraphs correspond to the music "Angola Bond", "Who is he", "Dangerous", "Pleasure", and "Strip". The regions of different colors correspond to different hyperedges. The hyperedges correspond to music styles in the left panel and to music tags in the right panel.

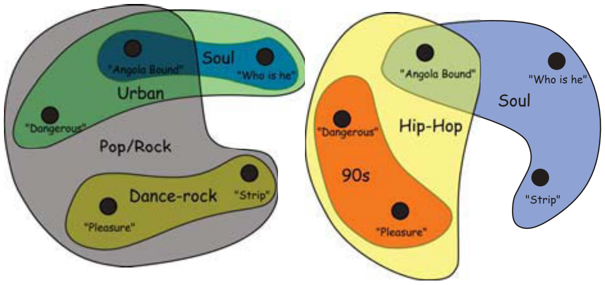


Figure 8: An example of the music style (left) and tag (right) hypergraph

Keep the concept of hypergraph in mind, and now let us describe our proposed multi-label classification algorithm:

Suppose there are n training samples $\{(x_i, y_i)\}_{i=1}^{n}$, where each instance x_i is drawn from some domain $\mathcal{X} \subseteq \mathbb{R}^m$ and its label y_i is a subset of the output label set $\mathcal{Y} = \{1, \cdots, k\}$. For example, if x_i belongs to categories 1, 3, and 4, then $y_i = \{1,3,4\}$. We use $\mathbf{X} = (x_1, \cdots, x_n)^T$ to represent the data feature matrix.

The basic strategy of this algorithm is to solve the multi-label learning by combing a label ranking problem and a label number prediction problem. That is, for each instance we produce a ranked list of all possible labels, estimate the number of labels for the instance, and then select the predicted number of labels from the list.

Label Ranking Algorithm

Label ranking is the task of inferring a total order over a predefined set of labels for each given instance [37]. Generally, for each category, we define a linear function $f_i(x) = \langle w_i, x \rangle + b_i \ \ (i = 1, \cdots, k)$, where $\langle \cdot, \cdot \rangle$ is the inner product and b_i is a bias term. One often deals with the bias term by appending to each instance an additional dimension

$$x^T \leftarrow [x^T, 1], \ \ w_i^T \leftarrow [w_i^T, b_i], \tag{7}$$

then the linear function becomes $f_i(x) = \langle w_i, x \rangle$. The goal of label ranking is to order $\{f_i(x), i = 1, \cdots, k\}$ for each instance x according to some predefined empirical loss function and complexity measures. Elisseeff and Weston [45] apply the large margin idea to multi-label learning and present an SVM-like ranking system, called Rank-SVM, given as follows:

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^{k} \|w_i\|^2 + C\sum_{i=1}^{n} \frac{1}{|y_i||\bar{y}_i|} \sum_{(p,q)\in y_i \times \bar{y}_i} \xi_{ipq} \\ \text{s.t.} \quad & \langle w_p - w_q, x_i \rangle \geq 1 - \xi_{ipq}, (p,q) \in y_i \times \bar{y}_i \\ & \xi_{ipq} \geq 0, \end{aligned} \tag{8}$$

where $C \geq 0$ is a penalty coefficient that trades off the empirical loss and model complexity, $\bar{y}_i$ is the complementary set of y_i in $\mathcal{Y}$, $|y_i|$ is the cardinality of the set y_i, i.e., the number of elements of the set y_i, and $\xi_{ipq}(i = 1, \cdots, n; (p,q) \in y_i \times \bar{y}_i)$ are slack variables. The margin term $\sum_{i=1}^{k} \|w_i\|^2$ controls the model complexity and improves the model generalization performance.

Although this approach performs better than Binary-SVM in many cases, it still does not model the category correlations clearly. Next, we will describe how to construct a hypergraph to exploit the category correlations and how to incorporate the hypergraph regularization into the problem in the form of Eq. (8).

To model the correlations among different categories effectively, a hypergraph is built where each vertex corresponds to one training instance and a hyperedge is constructed for each category which includes all the training instances relevant to the same category. Here, we apply the Clique Expansion algorithm [26] to construct the similarity matrix of the hypergraph. It means that the similarity of two instances is proportional to the sum of the weights of their common categories, thereby captures the higher order relations among different categories. This kind of hypergraph structure was used in the feature extraction by spectral learning [140]. However, we consider how to apply the relation information encoded in the hypergraph to directly design the multi-label learning model. Intuitively, two instances tend to have a large overlap in their assigned categories if they share high similarity in the hypergraph. Formally, this smoothness assumption can be expressed using the hypergraph Laplacian regularizer, trace$(\widehat{\mathbf{F}}^T\mathbf{L}\widehat{\mathbf{F}})$. Therefore we can introduce the smoothness assumption into Eq. (8) and obtain

$$\begin{aligned}
\min \quad & \frac{1}{2}\sum_{i=1}^{k}\|w_i\|^2+\frac{1}{2}\lambda\text{trace}(\widehat{\mathbf{F}}^T\mathbf{L}\widehat{\mathbf{F}})+\\
& C\sum_{i=1}^{n}\frac{1}{|y_i||\bar{y}_i|}\sum_{(p,q)\in y_i\times\bar{y}_i}\xi_{ipq}\\
\text{s.t.} \quad & \langle w_p-w_q,x_i\rangle\geq 1-\xi_{ipq},(p,q)\in y_i\times\bar{y}_i\\
& \xi_{ipq}\geq 0. \qquad (9)
\end{aligned}$$

Here $\widehat{\mathbf{F}}$ is the matrix of label prediction, that is, the $n\times k$ matrix $(f_j(x_i))$, $1\leq i\leq n$, $1\leq j\leq k$. $\mathbf{L}$ is the $n\times n$ hypergraph Laplacian and $\lambda\geqslant 0$ is a constant that controls the model complexity in the intrinsic geometry of input distribution.

Problem (9) is a linearly constrained quadratic convex optimization problem. To solve it, we first introduce a dual set of variables, one for each constraint, i.e., $\alpha_{ipq}\geq 0$ for $\langle w_p-w_q,x_i\rangle -$

$1+\xi_{ipq} \geq 0$ and η_{ipq} for $\xi_{ipq} \geq 0$. After some linear algebraic derivation, we obtain the dual of Problem (9) as

$$\begin{aligned}\min g(\alpha) &= \frac{1}{2}\sum_{p=1}^{k}\sum_{h,i=1}^{n}\beta_{ph}\beta_{pi}x_h^T(I+\lambda X^T LX)^{-1}x_i \\ &\quad -\sum_{i=1}^{n}\sum_{(p,q)\in y_i\times\bar{y}_i}\alpha_{ipq} \\ \text{s.t.} &\quad 0\leq\alpha_{ipq}\leq\frac{C}{|y_i||\bar{y}_i|}, \qquad (10)\end{aligned}$$

where α denotes the set of dual variables α_{ipq} and I is the $(m+1)\times(m+1)$ identity matrix.

Once the variables α_{ipq} that minimize $g(\alpha)$ are obtained, we can compute w_p by

$$w_p = (I+\lambda X^T LX)^{-1}\sum_{i=1}^{n}\beta_{pi}x_i, \qquad (11)$$

where

$$\beta_{pi} = \sum_{(j,q)\in y_i\times\bar{y}_i} t_{ijq}^{p}\alpha_{ijq} \qquad (12)$$

$$t_{ijq}^{p} = \begin{cases} 1 & j=p \\ -1 & q=p \\ 0 & \text{if } j\neq p \text{ and } q\neq p. \end{cases} \qquad (13)$$

Compared to the primal optimization problem, the dual has $k(m+1)$ less variables and includes simple box constraints. The dual can be solved by the dual coordinate descent algorithm shown in Algorithm 1.

Label Number Prediction Algorithm

To identify the final labels of data, we need to design an appropriate threshold for each instance to determine the size of its corresponding label set. Here, we adopt the strategy proposed by Elisseeff and Weston [45], which treats threshold designing as a supervised learning problem. More concretely, for each instance x, define a threshold function $h(x)$ and the

Algorithm 1 A dual coordinate descent method for HiSVM

```
Start with $\alpha = \mathbf{0} \in \mathbb{R}^{n_\alpha}$ ($n_\alpha = \sum_{i=1}^{n} |y_i||\bar{y}_i|$), and the corresponding $w_i = \mathbf{0}$ $(i = 1, \cdots, k)$
while 1 do
    for $i = 1, \cdots, n$ and $(j, q) \in y_i \times \bar{y}_i$ do
        1. $G = (w_p - w_q)^T x_i - 1$
        2. $PG = \begin{cases} G & \text{if } 0 < \alpha_{ipq} < \frac{C}{|y_i||\bar{y}_i|} \\ \min(0, G) & \text{if } \alpha_{ipq} = 0 \\ \max(0, G) & \text{if } \alpha_{ipq} = \frac{C}{|y_i||\bar{y}_i|} \end{cases}$
        3. If $|PG| \neq 0$,
            $\alpha^*_{ipq} \leftarrow \min\left(\max\left(\alpha_{ipq} - \frac{G}{2A_{ii}}, 0\right), \frac{C}{|y_i||\bar{y}_i|}\right)$
            $w_p \leftarrow w_p + (\alpha^*_{ipq} - \alpha_{ipq})(I + \lambda X^T L X)^{-1} x_i$
            $w_q \leftarrow w_q - (\alpha^*_{ipq} - \alpha_{ipq})(I + \lambda X^T L X)^{-1} x_i$
    end for
    if $\|\alpha^* - \alpha\| / \|\alpha\| < \varepsilon$(e.g. $\varepsilon = 0.01$) then
        Break
    end if
    $\alpha = \alpha^*$
end while
```

size of label set $s(x) = \|\{j \mid f_j(x) > h(x), j = 1, \cdots, k\}\|$. Our goal is to obtain $h(x)$ through a supervised learning method. For the training data x_i, its label ranking values, $f_1(x_i), \cdots, f_k(x_i)$, can be given by the foregoing ranking algorithm, and its corresponding threshold $h(x_i)$ is simply defined by

$$h(x_i) = \frac{1}{2}(\min_{j \in y_i}\{f_j(x_i)\} + \max_{j \in \bar{y}_i}\{f_j(x_i)\}).$$

Once the training data $(x_1, h(x_1)), \cdots, (x_u, h(x_u))$ are generated, we can use off-the-shelf learning methods to learn $h(x)$. In this study, Linear Support Vector Regression [148] has been adopted to solve $h(x)$. Note that all the label ranking based algorithms toward multi-label learning can use this postprocessing approach to predict the size of label set.

3.4.2 Description of Experiments

Dataset Description

For experimental purpose, we created a data set consisting of 403 artists. For each artist, we include a representative song and also obtain the style and tag description. Music audio data were provided by `http://www.newwisdom.net`. For experimental purpose, we created a common data set of 403 artists that we can find the music audio data and the mood, style, as well as tag descriptions. We requested 1 piece of audio data for each artist. As the songs in our test domain tend to have introductory non-vocal parts in the first 60 second, we generate a music sample using the third 30-second block (i.e., between time 1'00" and 1'30") for each song. The audio feature extraction is performed as described in Section 3.2. The social Tag data came from the popular music website `http://www.last.fm`. An open research data set is available to download at `http://blogs.sun.com/plamere/entry/open_research_the_data_lastfm`. It was collected by Paul Lamere, a researcher in Sun Labs during the spring of 2007. Music tags are descriptions given by visitors or music tag editors from the website to express their idea on the music artists, albums or songs. Tags can be as simple as a word or as complicated as a whole sentence. Popular tags are terms like *rock*, *black metal*, and *indie pop*. Long tags are like *I love you baby can I have some more*. They are not as formal as style or mood description created by music experts. But they give us ideas of how large population music listeners think about the music artists, music albums or songs. In order to understand how important and accurate a tag is when reflecting an artist, the frequencies of all the tags to describe the artists (tag counts) were also taken into consideration in the experiments.

Experimental Setup

For the data set of 403 artists, we use 70% of the data for training (282 pieces total), and the remaining 30% for testing (121 pieces total). Here, the five models used for multi-label learning are compared as follows:

- Binary-SVM. In this model, first, for each category, train a linear SVM classifier independently. Then, the labels for each test instance can be obtained by aggregating the classification results from all the binary classifiers. Here, we use LIBSVM [23] to train the linear SVM classifiers.

- Rank-SVM [45]. In this model, first, using Eq. (8), we implement the optimization algorithm [150] ($\lambda = 0$) to train a linear label ranking system. We then apply the prediction method for the size of label set described in Section 3.4.1 to design the threshold model. Finally, for each test instance, we combine the label ranking and threshold models, thereby infer its labels.

- HiSVM. This is our proposed algorithm. The algorithm is composed of three steps: (1) we implement the optimization algorithm [150] to achieve a linear label ranking system; (2) we apply the method in Section 3.4.1 to design the threshold model; (3) for each test instance, we combine the label ranking and threshold models to infer its labels.

- HSVM. HSVM is the style Hypergraph regularized SVM method, which is the same as the HiSVM method except that it only makes use of the style hypergraph and does not use the tag hypergraph.

- GSVM. GSVM is similar to HiSVM except we construct a traditional 2-graph where each vertex represents one training instance in GSVM rather than a hypergraph. In order to compute the Laplacian, the weight w_{ij} of the edge between x_i and x_j is defined as follows

$$w_{ij} = \exp(-\rho\|x_i - x_j\|^2), \tag{14}$$

where ρ is a nonnegative constant. Apparently, the category correlation information is not used during the construction of 2-graph in GSVM.

Experimental Results

Table 6 illustrates the experimental results on our HiSVM algorithm along with the four other methods on the data set. The values in Table 6 are the F_1 Micro values and F_1 Macro values averaged over 50 independent runs together with their standard deviations.

Methods	F1 *Macro*	F1 *Micro*
Binary-SVM	0.4231 ± 0.0025	0.4317 ± 0.0103
Rank-SVM	0.4526 ± 0.0114	0.4733 ± 0.0036
GSVM	0.5018 ± 0.0054	0.5244 ± 0.0103
HSVM	0.5365 ± 0.0120	0.5509 ± 0.0072
HiSVM	$\mathbf{0.5613 \pm 0.0069}$	$\mathbf{0.5802 \pm 0.0116}$

Table 6: Performance comparisons of four models on the last.fm dataset

From the table we can clearly observe the following:

- Multi-label methods perform better than the simple Binary-SVM method.

- The consideration of label correlations is helpful for the final algorithm performance.

- Hypergraph regularization is better than flat two-graph regularization because it can incorporate the high-order label relationships naturally.

- The incorporation of tag information is helpful for the final classification performance.

Figure 9 shows how the relative error $\|\alpha^* - \alpha\|/\|\alpha\|$ varies with the process of iteration using the dual coordinate descent method introduced in Algorithm 1.

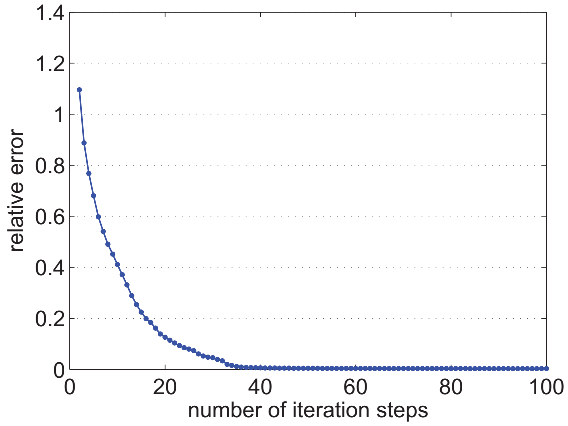


Figure 9: The relative error vs. iteration step plot of our proposed dual coordinate descent algorithm for solving HiSVM

From the figure we clearly see that with the process of coordinate descent, the relative error will decrease and it takes approximately 30 steps to converge. This validates the correctness of our algorithm experimentally.

CHAPTER 4

MUSIC ANALYSIS FROM DIFFERENT INFORMATION SOURCES

4.1 Introduction

As described in the previous chapter, the music data are naturally multi-modal, in the sense that they are represented by multiple sets of features. For example, the representation of a song can have four dimensions of features: 1) the personnel dimension, including the singer, the composer, the producer, the director, the editor, the scenario writer, the cast, and so on; 2) the lyric features; 3) the content features, which summarize the voice and background audio. They can be acoustic features extracted from audio recordings or higher-level features extracted from MIDI files [96]; and 4) the feedback from listeners or labels from music experts, such as tags, genres, mood and styles. Having data with heterogeneous sets of features, one may pose a natural question: can multi-modality be effectively utilized in music data analysis, and if so, can such multi-modal learning methods produce better analysis results than uni-modal methods?

Two fundamental approaches in dealing with the music data are classification and clustering. While classification assigns predefined class labels to the data, clustering divides the data into classes based on their similarity without predefined class labels. Since it requires user input (or labeled data) for training, the former approach is called *supervised learning*, while the latter approach does not require user input (or only use unlabeled data), and thus is called *unsupervised learning*. Practically, these approaches can be revised, while the data sets can be combined to improve the organization performance and accuracy. While there is a vast literature on music classification, the problem of music clustering is much less explored [27, 111, 143].

Note that many strategies such as co-learning [1, 13, 33, 123] and co-boosting [29] have been developed to perform supervised learning as well as semi-supervised learning (where both labeled and unlabeled data are used for training) from the data with heterogeneous sets of features. In our work, we study the issue of clustering pop music into groups with respect to the artists from diverse information sources.

We first develop a new bi-modal music clustering algorithm for integrating the features based on minimizing disagreement. To apply the bi-modal music clustering, we need to have a complete feature representation, i.e., we need to know the content and lyrics information for each song. However, sometimes we might not be able to get the complete feature representation. For example, the lyrics may not be available for certain songs in our study. In addition, in many cases, some data sources may not be as informative as other data sources. For example, the lyrics may not be able to provide the same level of details of genre/style information as the acoustic features. This motivates us to study music clustering with constraints: One data source is chosen as primary information source. The other data sources are treated as secondary information and are used as constraints to improve the clustering results based on the primary source.

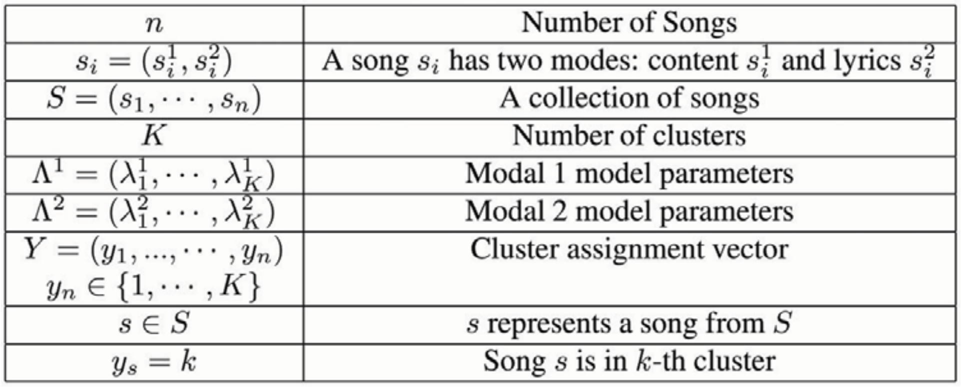
In summary, this chapter studies the following two related problems:

- Bi-modal music clustering: Note that in music information retrieval, the personnel feature set of the representation of music, is significantly smaller than that of movies, since many music artists produce, compose, and perform themselves. This compels one to take the standpoint that the representation of popular music is bimodal, consisting of the acoustic features, which summarize the sound, and the text features, which summarize the words put into the music. To apply the bi-modal music clustering, we need to have a complete feature representation, i.e., we need to know the content and lyrics information for each song. We of course anticipate that bimodal clustering techniques can be naturally extended to general multi-modal clustering.

- Music clustering with constraints: In practice, bi-modal clustering might not be plausible for the following two scenarios: (1) The feature set from some information source might not be sufficient enough to represent the music (e.g., the personnel features described above); (2) We may not always have the complete feature representation. For instance, sometimes we only have the lyrics information or meta-data information of a small number of songs. To utilize these partial or incomplete information from diverse information sources, we represent it as instance-level constraints (e.g., two artists share similar lyrics or personnel features) and study the problem of music clustering with constraints [115].

The rest of the chapter is organized as follows: Section 4.2 presents the bi-modal clustering algorithm, Section 4.3 describes constraint-based clustering algorithm, Section 4.4 introduces text-based feature extraction and constraint generation, Section 4.5 shows the experimental results, and finally Section 4.6 concludes the chapter.

4.2 Bimodal Clustering Algorithm Description

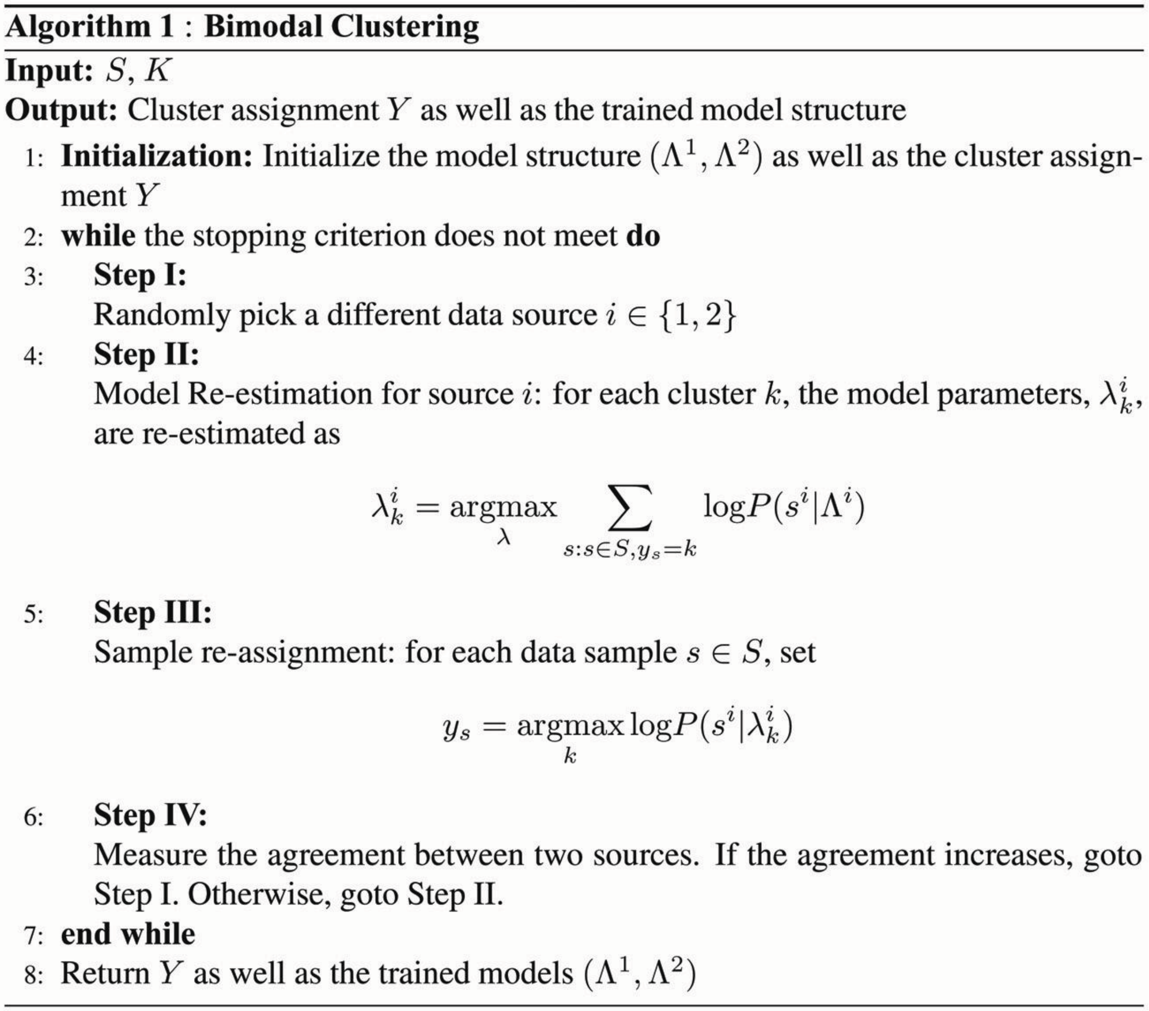


n	Number of Songs
$s_i = (s_i^1, s_i^2)$	A song s_i has two modes: content s_i^1 and lyrics s_i^2
$S = (s_1, \cdots, s_n)$	A collection of songs
K	Number of clusters
$\Lambda^1 = (\lambda_1^1, \cdots, \lambda_K^1)$	Modal 1 model parameters
$\Lambda^2 = (\lambda_1^2, \cdots, \lambda_K^2)$	Modal 2 model parameters
$Y = (y_1, ..., \cdots, y_n)$ $y_n \in \{1, \cdots, K\}$	Cluster assignment vector
$s \in S$	s represents a song from S
$y_s = k$	Song s is in k-th cluster

Table 7: The list of notations used in Bimodal Clustering Algorithm

Our clustering algorithm is based on the the basic principle of minimizing disagreement, which is claimed in [83]: minimizing the disagreement between two individual models could lead to the improvement of learning performance of individual models. It should be pointed out that although the principle of minimizing the disagreement was originally proved in the context of supervised learning [35], it can be regarded as a simple common theme of multi-modal information retrieval: individual feature sets interact to help each other by reducing disagreement among their outputs.

The clustering algorithm can be considered as an extension of the EM method [39]. In each iteration of the algorithm, an EM type procedure (an Expectation step followed by a



```
Algorithm 1 : Bimodal Clustering
Input: $S$, $K$
Output: Cluster assignment $Y$ as well as the trained model structure
Initialization: Initialize the model structure $(\Lambda^1, \Lambda^2)$ as well as the cluster assignment $Y$
while the stopping criterion does not meet do
    Step I:
    Randomly pick a different data source $i \in \{1, 2\}$
    Step II:
    Model Re-estimation for source $i$: for each cluster $k$, the model parameters, $\lambda_k^i$, are re-estimated as
        $\lambda_k^i = \underset{\lambda}{\text{argmax}} \sum_{s: s \in S, y_s = k} \log P(s^i | \Lambda^i)$
    Step III:
    Sample re-assignment: for each data sample $s \in S$, set
        $y_s = \underset{k}{\text{argmax}} \log P(s^i | \lambda_k^i)$
    Step IV:
    Measure the agreement between two sources. If the agreement increases, goto Step I. Otherwise, goto Step II.
end while
Return $Y$ as well as the trained models $(\Lambda^1, \Lambda^2)$
```

Figure 10: Bimodal Clustering Algorithm

Maximization step) is employed to bootstrap the model by starting with the cluster assignments obtained in the previous iteration. At the end of each iteration, the algorithm explicitly checks whether the agreement between two clusterings (one clustering from each data source) has been improved. If it is improved, the algorithm continues to iterate. Otherwise, it will go back to the allocation step and try to get a new clustering. Table 7 lists the notions used for the bimodal clustering Algorithm while Figure 10 gives a formal description of the algorithm procedure.

4.3 Constraint-based Clustering

In our work, music clustering with constraints [115] is also studied. In practice, bimodal clustering might not be feasible in the following situations: 1) The feature set from some information source is not sufficient to represent the music; 2) We do not always have the complete feature representation. For instance, sometimes we only have the lyrics information or meta-data information of a limited number of songs. In such cases, the incomplete or partial information are just used as constraints to improve the clustering performance. This section provides some background on the K-means algorithm and then discusses the constraint-based clustering algorithm following the exposition in [34].

4.3.1 K-means Clustering

K-means is a popular clustering algorithm where the input data set is partitioned into K groups, where the number K is specified by the user. The quality of partition into K clusters can be viewed as the *quantization error* described below:

$$E = \frac{1}{2}\sum_{j=1}^{K}\sum_{s\in C_j}(\bar{c}_j - s)^2. \tag{15}$$

Here $C_1,\ldots,C_K$ are the K clusters and $\bar{c}_1,\ldots,\bar{c}_K$ their centroids. The goal of K-means is to minimize this quantization error, which is accomplished by iteratively alternating between the *allocation step* and the *evaluation step.* In the former each data point is allocated to the cluster whose centroid is the closest to it so as to minimize the quantization error with respect to the current centroids, while in the latter, the centroid of each cluster is updated based on the new allocation.

4.3.2 Constraint-based Clustering

Following [34] we define the concept of constraint-based clustering for music similarity. We modify the the objective function so that penalty is added for each constraint that is not satisfied. For a positive constraint (s_i, s_j) the penalty (in the case where they go to different clusters) is the squared distance between their cluster centroids. For a negative constraint (s_i, s_j) the penalty (in the case where they go to the same clusters) is the squared distance between the centroids that are the closest and the second closest to either s_i or s_j. In both cases, we use the centroids to determine the penalty so as to treat constraint violations equally within a cluster, and we use squared distance since the quantization error is based on squared distance.

The formula for the objective function is given bellow:

$$\begin{aligned}
CE &= \frac{1}{2}(E+PM+PC) \qquad (16)\\
&= \frac{1}{2}\left(\sum_{j=1}^{K}\sum_{s\in C_j}(\bar{c}_j - s)^2 + PM + PC\right) \qquad (17)\\
PM &= \sum_{(s_i,s_j)\in M} p_{ij}^m (1-\Delta(y(s_i),y(s_j))), \qquad (18)\\
PC &= \sum_{(s_i,s_j)\in C} p_{ij}^c \Delta(y(s_i),y(s_j)), \qquad (19)\\
p_{ij}^m &= (\bar{c}_{y(s_i)} - \bar{c}_{y(s_j)})^2, \qquad (20)\\
p_{ij}^c &= (\bar{c}_{y(s_i)} - \bar{c}_{ij}^{*})^2. \qquad (21)
\end{aligned}$$

Here M and C respectively represent the set of positive constraints and the set of negative constraints, p_{ij}^m and p_{ij}^c are respectively penalty parameters for the positive and negative constraints, and the value of $y(s_i)$ is the index of the cluster to which the data point s_i belongs. Also, Δ is the Kronecker delta function defined by: $\Delta(x,y) = 1$ if $x = y$ and 0 otherwise. That is, the penalty p_{ij}^m is added only if $(s_i,s_j) \in M$ but s_i and s_j belong to different clusters; and the penalty p_{ij}^c is added only if $(s_i,s_j) \in C$ but s_i and s_j belong to the same cluster. Furthermore, $\bar{c}_{ij}^{*}$ is the centroid that is the next closest to either s_i or s_j.

Like K-means, the constraint-based clustering algorithm is iterative, alternating between the allocation step and the centroid update step. In the allocation step, the goal is to minimize the generalized constrained vector quantization error in Eq. 17 . This is achieved by assigning instances so as to minimize the proposed error term. For pairs of instances in the constraint set, the quantization error CE is calculated for each possible combination of cluster assignments, and the instances are assigned to the clusters so that CE is minimized. In the update step, the centroids are cluster centroids. As in K-means, the first order partial derivatives of CE with respect to each centroid is evaluated and the solution that makes all these derivatives equal to zero is obtained.

4.4 Feature Extraction and Constraints Generation

4.4.1 Lyrics-based Feature Extraction

In our work, we use the feature sets extracted from the lyrics and the acoustic content. The audio feature extraction is described in Chapter 3. To accommodate the characteristics of the lyrics, our text-based feature extraction consists of four components: bag-of-words features, Part-of-Speech statistics, lexical features and orthographic features.

- Bag-of-words: We compute the TF-IDF measure for each word and select top 200 words as our features. Stemming operations are not applied.

- Part-of-Speech statistics: We use the output of the part-of-speech (POS) tagger by Brill [12] as the basis for feature extraction. The POS statistics usually reflect the characteristics of writing. There are 36 POS features extracted from each document, one for each POS tag expressed as a percentage of the total number of words for the document.

- Lexical features: By "lexical features" we mean the features of individual word tokens in the text. The most basic lexical features are lists of 303 generic function words taken from [101], which generally serve as proxies for choice in syntactic (e.g., preposition phrase modifiers vs. adjectives or adverbs), semantic (e.g., usage of passive voice indicated by auxiliary verbs), and pragmatic (e.g., first-person pronouns indicating personalization of a text) planes. Function words have been shown to be effective style markers.

- Orthographic features: We also use orthographic features of lexical items, such as capitalization, word placement, word length distribution as our features. Word orders and lengths are very useful since the writing of lyrics usually follows certain melody.

4.4.2 Constraints Generation

The constraints come naturally in the context of music applications. Constraints can be generated from the background knowledge. If we already know that two songs are of the same styles, or formally, if we know two songs have the same cluster labels, then they must be in the same cluster (e.g., a positive constraint). Similarly, if it is known that two songs are of different styles, then they should be in different clusters (e.g., a negative constraint).

In our study, constraints can be generated from complementary and diverse music information sources. For example, if two piece of music have the same personnel-related features or lyrics, then they can be considered to be similar based on content.

4.5 Description of Experiments

4.5.1 Data Description

Our experiments are performed on the data set consisting of 570 songs from 53 albums of a total of 41 artists. The related audio recordings and the lyrics are collected. Acoustic features and lyrics-based features are then extracted using the approaches described above. In order to obtain the ground truth of song styles, we decided to use artist similarity information available at All Music Guide artist pages (http://www.allmusic.com), assuming that this information is the unbiased reflection of multiple individual users. On All Music Guide artist pages, if the name of an artist X appears on the list of artists similar to Y, we consider that X is similar to Y. The similarity graph of the 41 artists is shown in Figure 11. Following this direction, we identified four clusters for these 41 artists in our collection as listed in table 8. Our goal is to identify the song styles of the 570 songs in our data set using both the acoustic features and the lyrics-based features extracted.

Clusters	Members
No. 1	{Fleetwood Mac, yes, Utopia, Elton John, Genesis, Steely Dan, Peter Gabriel}
No. 2	{Carly Simon, Joni Mitchell, James Taylor, Suzanne Vega, Ricky Lee Jones, Simon & Garfunkel}
No. 3	{AC/DC, Black Sabbath, ZZ Top, Led Zeppelin, Grand Funk Railroad, Derek & The Dominos}
No. 4	All the remaining artists

Table 8: Cluster memberships

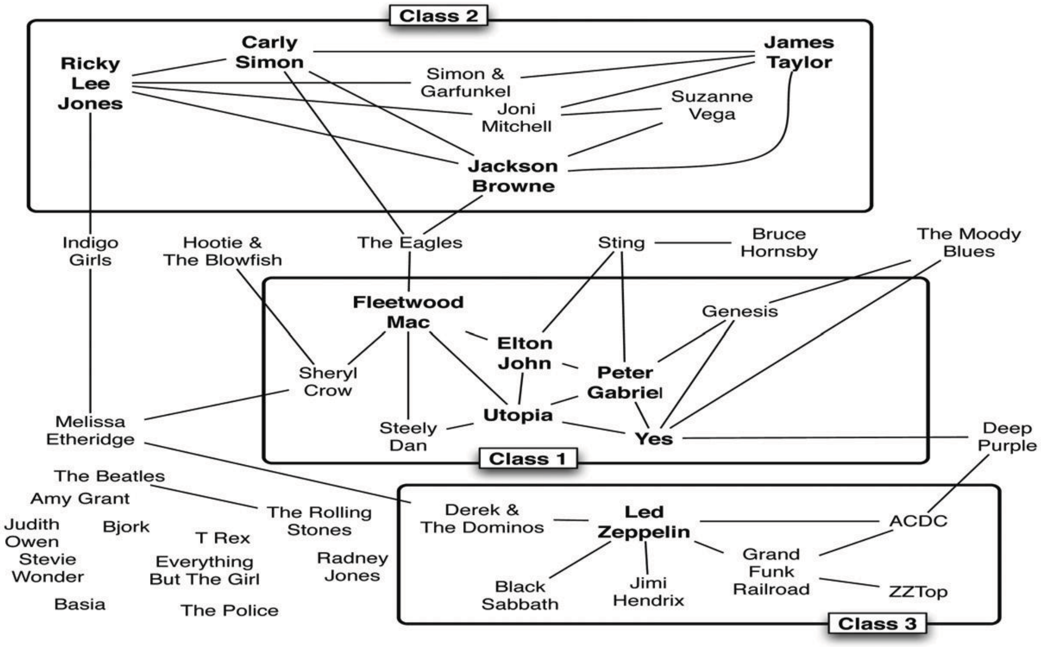


Figure 11: The artist similarity graph. The names in bold are “core” nodes.

4.5.2 Performance Measurement Criteria

We use Purity and Accuracy [42,162] as our performance measures of the clustering results. Purity measures the extent to which each cluster contains data points from primarily one class [162]. In general, the larger the purity value, the better the clustering solution is. Accuracy discovers the one-to-one relationship between clusters and classes, therefore measures the extent to which each cluster contains data points from the corresponding class [42]. It sums up the whole matching degree between all pair class-clusters. The larger accuracy value usually indicates the better clustering performance.

Feature Set(s)	Purity	Accuracy
Content-only	0.436	0.438
Lyrics-only	0.444	0.402
Feature-Level Integration	0.425	0.380
Cluster Integration	0.465	0.423
Sequential Integration I	0.431	0.434
Sequential Integration II	0.438	0.407
Bimodal Clustering	**0.471**	**0.453**

Table 9: Performance comparison for bimodal clustering. The numbers are obtained by averaging over ten trials.

4.5.3 Bi-modal Clustering Results

We compare the results of bimodal clustering with the results obtained when clustering is applied on content and lyrics separately, and with the results of other integration strategies. Table 9 presents the experimental results. From the table, we observe the following:

- The performance of purity and accuracy relative to the other is not always consistent in our comparison, i.e., higher purity values do not necessarily correspond to higher accuracy values. This is due to the fact that different evaluation measures consider different aspects of the clustering results.

- The purity and accuracy of feature-level integration are worse than those of content-only and lyric-only clustering methods. This shows that even though the joint feature space is more informative than that available from individual sources, naive feature integration tends to generalize the information poorly [155].

- Cluster Integration: Cluster integration refers to the procedure of obtaining a combined clustering from multiple clusterings of a data set [52, 102, 139]. Formally, let $C_1^1,...,C_1^{k_1}$ denote the clusters obtained from source 1, $C_2^1,...,C_2^{k_2}$ denote the clusters obtained from source 2. Each point d_i can then be represented as a $(k_1 + k_2)$-dimensional vector

$$d_i = (d_{i_1 1}, ..., d_{i_1 k_1}, ..., d_{i_2 1}, ..., d_{i_2 k_2})$$

$$d_{ijl} = \begin{cases} 1 & d_i \in C_j^{k_j} \\ 0 & otherwise, \end{cases} \quad for \ 1 \leq j \leq 2.$$

  A combined clustering can be found by applying the K-means algorithm on the new representation. The cluster integration performs better than content-only and lyrics-only. We can observe that cluster integration has higher purity and accuracy values than those of content-only and lyrics-only.

- Sequential Integration: Sequential integration is an intermediate approach of combining different information sources. It first performs clustering on one data source and obtains a clustering assignment, say, $C^1,...,C^{k_1}$. And each point d_i is represented as a k_i-dimensional vector using the similar idea in cluster integration. Then it combines the new representation with another data source using feature integration. Clustering can thus performed on the new concatenated vectors. Depending on the order of the two sources, we have two sequential integration strategies:

  - Sequential Integration I: firstly cluster based on content, then integrate with lyrics;
  - Sequential Integration II: firstly cluster based on lyrics, then integrate with content.

  The results of sequential integration are generally better than feature-level integration, and they are comparable with those of content-only and lyrics-only.

- Our bimodal clustering outperforms all other methods in all three performance measures.

The bimodal clustering algorithm can be regarded as a type of semantic integration of data from different information sources. The performance improvements proves that our bimodal clustering has advantages over the cluster integration. The bimodal clustering aims to minimize the disagreements between different sources and it can implicitly learn the correlation structure between different sets of features.

4.5.4 Experimental Results on Constraint-based Clustering

30 constraints (including 10 positive constraints and 20 negative constraints) are randomly generated from the cluster labels. We compare the results of constraint clustering with the results obtained when clustering is applied on content without any constraints. Table 10 presents the experimental results over ten independent trials.

Measurement	Purity	Accuracy
Without Constraints	0.436	0.438
With Constraints	0.471	0.472

Table 10: Performance comparison for clustering with constraints. The numbers are obtained by averaging over ten trials.

We observe that constraint-based clustering achieves better performance (i.e., higher purity and accuracy values) than clustering without any constraints, and that the performance of purity and accuracy relative to the other is consistent in our comparison, i.e., higher purity values correspond to higher accuracy values. Note that different evaluation measures consider different aspects of the clustering results. We hope that these different measures would provide enough information to understand the results of our experiments.

Figure 12 illustrates the effects of the constraint size. The X-axis of figure shows the number of constraints while the Y-axis shows the clustering accuracy. Here different constraint sizes are tested to investigate the effect of the size of the constraint on the overall clustering performance. An approximate 1 : 2 ratio of the number of positive constraints to the number of negative constraints is maintained throughout the experiment. We observe that as the constraint set size increases, the accuracy measures steadily improves and flattens out after 40. Then, after that, it looks as if the accuracy was to decrease. This may suggest that too many constraints may force our clustering algorithm to over-fit.

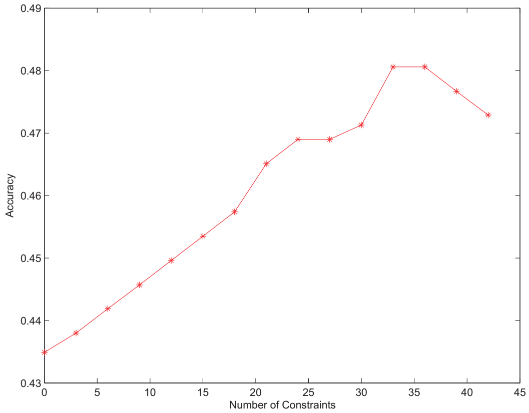


Figure 12: Comparisons of the clustering accuracy as a function of constraint size

4.6 Conclusion

In this chapter, we study the problem on whether multi-modal interactive methods can be more powerful than uni-modal methods in the case of clustering. In particular, we present a bi-clustering framework for integrating the features based on minimizing disagreement, and also provide a constraint-based clustering framework for clustering music songs in the presence of constraints. Experimental results show the effectiveness of our approaches.

CHAPTER 5

MUSIC RECOMMENDATION BASED ON ACOUSTIC FEATURES AND USER ACCESS PATTERNS

5.1 Introduction

5.1.1 Music Recommendation

Music recommendation is receiving increasing attention as the music industry develops venues to deliver music over the Internet. It is the procedure of providing a music listener a list of music pieces that he/she is likely to enjoy listening to. When the music data are well organized, annotated and analyzed using the strategies described in the previous chapters, music recommendation goal can be better reached. However, as we are intended to build a user-centric music information retrieval system, music recommendation should be based on a good understanding of the user preferences and the music pieces in the collection. Therefore, the key to a success music recommendation is to develop a good measurement strategy of the music similarity and an effective recommendation method based on the similarity measurement that can take the user preferences into account. Our goal for the music recommendation is to satisfy the following two requirements:

- *High recommendation accuracy.* A good recommendation system should output a relatively short list of songs in which many pieces are favored by the user and few pieces are not.

- *High recommendation novelty.* Good novelty is defined as rich artist variety and well-balanced music content variety. Music content represents the information of genre, timbre, pitch, rhythm, and so on [146]. Well-balance means that the music content is diverse and informative while not diverging much from the user's preferences.

Various music recommendation approaches have been developed in the literature, and they can be generally divided into two groups: collaborative-filtering methods and content-based methods. As discussed in chapter 2, both approaches have their own disadvantages: collaborative-filtering methods need a large collection of user history data and content-based methods lack the ability of understanding the interests and preferences of users. Probabilistic models and hybrid algorithms proposed recently also degraded significantly when they were short of corresponding user access data as illustrated in our experiments later in this chapter.

5.1.2 Contributions of this work

This chapter proposes a music recommendation approach by incorporating collaborative-filtering and acoustic contents of music. This approach employs a novel dynamic music similarity measurement strategy, which significantly improves the similarity measurement accuracy and efficiency. This measurement strategy utilizes the user access patterns from large numbers of users and represents music similarity with an undirected graph. Recommendation is calculated using the graph Laplacian and label propagation defined over the graph.

Figure 13 shows the framework of our proposed music recommendation system. First music data and user access patterns are collected and pre-processed. Then dynamic music similarity measurement is then used to compute the similarities between pairs of songs and construct the song graph. Finally, when seed songs are given, label propagation and ranking are performed for music recommendation. In the rest of the chapter, we call our recommendation approach as DWA since it utilizes dynamic weighting scheme based on user access patterns.

The proposed DWA approach is tested through experiments on a real data set constructed by anonymous users at `http://www.newwisdom.net` and has been adopted for music recommendation on that website.

5.2 Dynamic Music Similarity Measurement

5.2.1 Audio Similarity

Extraction of audio features for music similarity search has been well studied in the literature [49, 85, 91]. The use of acoustic features is justified by the fact that similar music

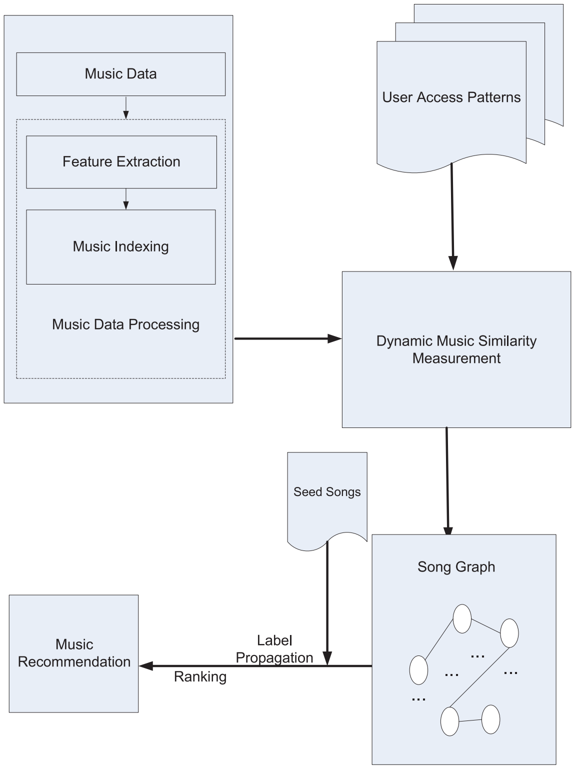


Figure 13: The framework of the proposed music recommendation approach

pieces use similar instruments and possess similar sound textures [43].

The music features are vectors in a multi-dimensional space, and the distance between the representation vectors characterizes and quantifies the closeness between two pieces of music. Traditionally there are two popular distance functions for measuring similarity in multimedia retrieval [48, 91, 124]:

- *Minkowski Distance Function.* Given two music songs A and B. Suppose their audio representations are given by two m dimensional vectors $(a_1, a_2, \cdots, a_m)$ and $(b_1, b_2, \cdots, b_m)$, respectively. The Minkowski distance $d(A,B)$ is then

  $$d(A,B) = \left(\sum_{i=1}^{m} |a_i - b_i|^p \right)^{\frac{1}{p}},$$

  where p is the Minkowski factor for the norm. In particular, if $p = 1$, this is the Manhattan distance, and if $p = 2$, it is the well-known Euclidean distance. The assumption of using Minkowski distance function is that the similar objects should be close in all dimensions as all the dimensions are treated equally.

- *Weighted Minkowski Distance Function.* The basic idea of weighted Minkowski distance function is to introduce weights to identify important features. If we assign each feature a weight w_i, then the weighted Minkowski distance function is

$$d_w(A,B) = \left(\sum_{i=1}^{m} w_i |a_i - b_i|^p \right)^{\frac{1}{p}}.$$

  The weighted Minkowski distance function is based on the static weighting scheme that assumes similar songs should be close in the same way (w.r.t to the same set of weights).

Although both distance functions have been previously used in music retrieval, they have the following two drawbacks:

- *Uniform weights for acoustic features.* In the Minkowski distance measurement, every audio feature is assigned with the equal weight when determining the similarity of music. This could be inappropriate given that people might be more sensitive to certain acoustic features than the others. This problem is further complicated when feature weights vary from one type of music to another. For example, for Rock, the audio intensity is an important feature in determining music similarity while it becomes a much less important feature for classic music. Thus, it is essential to assign dynamic weights to different acoustic features.

- *Subjective perception of music.* It is well known that the perception of music is subjective to individual users. Different users can have totally different opinions for the same pieces of music. Using a fixed set of weights for acoustic features is likely to fail in accounting for the taste of individual users. It is thus important to assign different weights to audio features based on the taste of individual users.

To address the above two issues, we propose a novel dynamic similarity measurement scheme. This scheme utilizes the access patterns of music from a considerable number of users.

	m_1	m_2	m_3	m_4
u_1	1	1	0	0
u_2	1	1	0	0
u_3	0	0	1	1
u_4	0	0	1	1

Table 11: An example of user access patterns

It is based on the assumption that two music pieces are similar in human perception when they share similar access patterns across multiple users. Table 11 illustrates the assumption. It shows a toy example of user access patterns on four pieces of music by four users. In this table, 1 represents that the music piece is accessed by the corresponding user while 0 indicates not. It is clear that m_1 and m_2 are similar from the user's viewpoint because they are accessed by users u_1 and u_2, but not by users u_3 and u_4. Also, m_3 and m_4 are similar to each other in that they are accessed by users u_3 and u_4, but not by u_1 and u_2. Similar ideas have been successfully applied to image retrieval to improve the accuracy of similarity measurement [55, 56, 104].

5.2.2 Dynamic Weighting Schemes

A simple approach capable of combining acoustic features and user access patterns for similarity measurement is to compute the similarity based on each representation and then combine the two similarity measurements linearly. By incorporating the user access patterns, the combined similarity measurement can more accurately reflect human perception of music than the one based only on acoustic features. A major drawback with such an approach is that user access patterns are usually sparse. Only for a relatively small number of music pieces, their user access data are adequate to provide robust estimation of similarity with other music pieces. This drawback will substantially limit the impact of the use of user access patterns. Also, since the approach uses the Minkowski distance for the audio-based similarity calculation, it cannot provide a means for estimating the weights on acoustic features, the essential components in making similarity measurement that is both genre-dependent and user-dependent.

Problem Formulation

Thus, the calculation of appropriate similarity measures can be casted as a learning problem aiming to assign approximate weights to each feature [152]. To automatically determine the weights for audio features, the metric learning approach [56, 156], which learns appropriate similarity metrics based on the correlation between acoustic features and user access patterns of music, needs to be explored. Given that human perception of music is well approximated by its user access patterns, a good weighting scheme for acoustic features should lead to a similarity measurement that is consistent with the one based on user access patterns. Let $m_i = (\mathbf{a}_i, \mathbf{u}_i)$ denote the i-th piece of music in the data set, where $\mathbf{a}_i$ and $\mathbf{u}_i$ represent its acoustic features and user access patterns, respectively. Let $S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) = \sum_l a_{i,l} a_{j,l} w_l$ be the sound-based similarity measurement between the i-th and the j-th pieces of music when the parameterized weights are given by $\mathbf{w}$. Let $S_u(\mathbf{u}_i, \mathbf{u}_j) = \sum_k u_{i,k} u_{j,k}$ be the similarity measurement between the i-th and j-th pieces of music based on their user access patterns. Here for each k, $u_{i,k}$ denotes whether the k-th user accesses the i-th piece of music. To learn appropriate weights $\mathbf{w}$ for audio features, we can enforce the consistency between similarity measurements $S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w})$ and $S_u(\mathbf{u}_i, \mathbf{u}_j)$. The above idea leads to the following optimization problem:

$$\begin{aligned} \mathbf{w}^* &= \arg\min \sum_{i \neq j} (S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) - S_u(\mathbf{u}_i, \mathbf{u}_j))^2 \\ \text{s.t.} &\quad \mathbf{w} \geq 0. \end{aligned} \tag{22}$$

Let p be the number of content features. The summation in Equation 22 is rewritten as follows:

$$\begin{aligned} &\sum_{i \neq j} (S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) - S_u(\mathbf{u}_i, \mathbf{u}_j))^2 \\ &\quad = \sum_{i \neq j} (a_{i,1} a_{j,1} w_1 + \cdots + a_{i,p} a_{j,p} w_p - \sum_k u_{i,k} u_{j,k})^2 \\ &\quad = \sum_{i \neq j} ((a_{i,1} a_{j,1} w_1 + \cdots + a_{i,p} a_{j,p} w_p)^2 - 2(a_{i,1} a_{j,1} w_1 + \\ &\qquad \cdots + a_{i,p} a_{j,p} w_p)(\sum_k u_{i,k} u_{j,k}) + (\sum_k u_{i,k} u_{j,k})^2), \end{aligned}$$

where $a_{i,l}$ is l-th feature in the acoustic feature set a_i and $a_{j,l}$ is l-th feature in the acoustic feature set a_j. Let n be the number of pieces of music, and let

$$A = \begin{bmatrix} a_{1,1}a_{2,1} & a_{1,2}a_{2,2} & \dots & a_{1,f}a_{2,f} \\ & \dots & \dots & \\ a_{n-1,1}a_{n,1} & a_{n-1,2}a_{n,2} & \dots & a_{n-1,f}a_{n,f} \end{bmatrix}$$

and

$$U = \begin{bmatrix} \sum_{i \neq j} a_{i,1}a_{j,1}(\sum_k u_{i,k}u_{j,k}) \\ \vdots \\ \sum_{i \neq j} a_{i,f}a_{j,f}(\sum_k u_{i,k}u_{j,k}) \end{bmatrix},$$

where A is a $(C_n^2 \times p)$ matrix and U an $(p \times 1)$ matrix. Thus, Equation 22 is equivalent to:

$$\begin{aligned} \mathbf{w}^* &= \arg\min \left[\frac{1}{2} \times 2(Aw)^T(Aw) - U^T w \right] \\ &= \arg\min \left[\frac{1}{2} \left(w^T(2A^TA)w + (-2U^T)w \right) \right] \\ \text{s.t.} \quad & \mathbf{w} \geq 0. \end{aligned} \tag{23}$$

This optimization problem can be addressed using quadratic programming techniques [51].

Discussions

A similar strategy can be applied to make the similarity measurement dependent on the preferences of individual users. This is accomplished by selecting a subset of users whose access patterns are similar to those of the active users and then use only those selected in the estimation of music similarity. In other words, the quantity $S_u(\mathbf{u}_i, \mathbf{u}_j)$ in Equation 22 is estimated only based on those users that are deemed similar. An important issue in employing such an approach is the method and the cost of selecting similar users. One possibility is to use the min-wise hash indexing scheme (to be discussed in Section 5.3.1), in which a set of t independent hash functions are applied to each component of the user access pattern vector, which is of dimension n and the minimum of the t values is chosen as the hash value of each component. Then two representations are compared for similarity by simply counting how many components have the same hash value. By applying a simple threshold to the count, similar users can be selected. The time that it takes to compute similarity is $O(n)$ for each pair of users, assuming that the hash values have been already computed. Therefore, the selection of similar users to the active user requires time $O(nm)$, where m is the number of users. This possibility is not explored here in this work since the number m of the dataset is small.

5.3 Music Recommendation Over Song Graph

In this study, we employ timbral features and wavelet coefficient histograms for feature extraction. The extracted feature set consists of the following three components and total 80 features. The detailed process has been described in the previous chapter and is omitted here.

5.3.1 Music Indexing

Once the features/signatures for each song are obtained, efficient data structures can be built for similarity search. In this study, min-wise hashing [15] is used to speed up similarity computation for large data sets, especially in online calculation. The key idea is that we can create a small signature for each song and the resemblance of any pair of songs s_i and s_j can be accurately estimated based on their min-wise hashing signatures.

The min-wise hashing signature is computed as follows. Given a signature of size r, r independent random hash functions $f_1, \ldots, f_r$ are firstly generated. For a song s_i (s_i is the feature set of song i), the t-th component of its signature is given by

$$\min\{f_t(d) \mid d \in s_i\},$$

where d represents any feature in the feature set.

In doing so, the minimal hash value in s_i for the t-th hash function f_t is reserved. Note that the same hash function f_t is used for every song to generate its t-th signature component. Let S^i and S^j be the signatures of s_i and of s_j thus obtained, respectively. Let S^i_t and S^j_t be the t-th components of S^i and S^j. We say that they match at t if $S^i_t = S^j_t$. The resemblance between s_i and s_j can be then measured by the proportion of the number of matches between S^i and S^j to r, the number of components.

The min-wise hashing estimator is unbiased. An error bound was given in [15] and the accuracy increases with the resemblance value. Note that the number of matches between two signatures can be computed in $O(r)$ time and that r is independent of the size of database.

5.3.2 Song Graph

In previous section, we presented an efficient method to compute the similarities between pairs of songs. We are now ready to construct the song graph.

Definition 1 (Song graph). *A song graph is an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where*

1. *$\mathcal{V} = \mathcal{I}$ is the node set ($\mathcal{I}$ is the song set, which means that each song is represented as a node on the graph $\mathcal{G}$);*

2. *$\mathcal{E}$ is the edge set. Associated with each edge $e_{pq} \in \mathcal{E}$ is the similarity w_{pq}, which is nonnegative and satisfies $w_{pq} = w_{qp}$.*

Once the song graph is constructed, music recommendation can be treated as a label propagation from labeled data (i.e., items with ratings) to unlabeled data. In its simplest form, the label propagation is like a random walk on a song graph $\mathcal{G}$ [141]. Using diffusion kernel [71, 138], the label propagation is like a diffusive process of the labeled information [163,165]. Zhu et al. [165] utilizes the harmonic nature of the diffusive function, Zhou et al. [163] emphasize the spread of label information in a consistent and iterative way. Motivated from the previous research, we emphasize the global and coherent nature of label propagation and use the Green's function of the Laplace operator for music recommendation [41].

5.3.3 Label Propagation on Graph

Given a graph with edge weights T, the *combinatorial Laplacian* is defined to be $L = D - T$, where D is the diagonal matrix consisting of the row sums of W; i.e., $D = \mathrm{diag}(T\mathbf{e})$, $\mathbf{e} = (1 \cdots 1)^T$.

Green's function is defined on the generalized eigenvectors of the Laplacian matrix:

$$L\mathbf{v}_k = \zeta_k D\mathbf{v}_k, \;\; \mathbf{v}_p^T D\mathbf{v}_q = \mathbf{z}_p^T \mathbf{z}_q = \delta_{pq}, \tag{24}$$

where $0 = \zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_n$ are the eigenvalues and the zero-mode is the first eigenvector $\mathbf{v}_1 = \mathbf{e}/\sqrt{n}$. Then we have

$$G = \frac{1}{(D-T)_+} = \sum_{k=2}^{n} \frac{\mathbf{v}_k \mathbf{v}_k^T}{\zeta_k}. \tag{25}$$

In practice, the expansion after some K terms is truncated and the K vectors are stored. Green's function is computed on the fly. Therefore the storage requirement is $O(Kn)$.

The recommendation on the song graph is illustrated in Figure 14. In the figure, the colored (shaded) nodes represent the rated items with their corresponding ratings. The others are the unrated items, whose ratings are unknown. Let $\mathbf{y}^T = (y_1, \cdots, y_n)$ be the rating for a user. Given an incomplete rating $\mathbf{y}_0^T = (5, ?, ?, 4, 2, ?, ?, ?, 3)$, the question is to predict those missing values. Using Green's function, we initialize $\mathbf{y}_0^T = (5, 0, 0, 4, 2, 0, 0, 0, 3)$, and then compute the complete rating as the linear influence propagation

$$\mathbf{y} = G\mathbf{y}_0, \tag{26}$$

where G is the Green function built from the song graph.

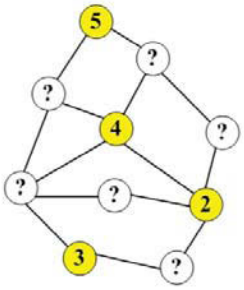


Figure 14: An illustration of a recommendation task

5.3.4 Music Ranking

After label propagation, the ratings for unrated songs are obtained and many of them might have the same rating. In practice, a ranked list of the items to be recommended is required. The music ranking over a song graph $\mathcal{G}$ can be treated as the problem of finding the shortest path from the seed song node to the rest of the nodes in the song graph. The edges with low similarity have already been eliminated, so only the remaining edges can be used to construct shortest paths. For any $M \geq 1$, to recommend M songs after a seed song s, we simply select the M songs that are the closest to s. The standard single-source shortest-path algorithm produces the shortest path to any node in time $O(|\mathcal{V}|^2 + |\mathcal{E}| \log |\mathcal{V}|)$ where $|\mathcal{V}|$ is the number of nodes and $|\mathcal{E}|$ is the number of edges in the graph. The time that it takes for identifying M closest nodes after the shortest path length is obtained can be $O(M|\mathcal{V}|)$.

5.4 Experiments and Evaluation

In this section, we present the performance evaluation of our music recommendation system, including effectiveness and novelty analysis. Various case studies and the user study show the promising recommendation quality of our system.

5.4.1 Data Collection

The music data were collected from `http://www.newwisdom.net`. It is a website in Chinese language with major functions of education and entertainment. This website has approximately 10,000 registered users visiting its forums regularly. These users also listen to music and meanwhile create their own favorite playlists (called CDs on this website). The website had a collection of more than 10,000 songs and hundreds of playlists at the moment of this experimentation. More than 80% of songs were from famous Chinese artists, others were from famous American, European, Japanese, and Korean artists. The songs covered many different genres including Pop, Classical, Jazz, Rock, Country, R&B, Blues, Disco, Rap and Hip-hop.

In the experiments described below, we sampled 2829 songs from the playlists created by “serious” users in the same group on the website. The criterion for a “serious” user is the number of songs in his/her playlists. We eliminated those whose playlists containing either less than 10 or more than 20 songs from the data collection. Those users are assumed to be either “too uninterested” or “too eager.” and then defined not “serious”. This culling process leaves us 274 playlists.

5.4.2 Data Processing

We process the collected songs and user playlists to get the content features and user access patterns. Then our dynamic weighting scheme and music ranking algorithm are applied to generate the recommendation identifications of music pieces.

Acoustic Feature Representation

For each song, a music sample using the third 30-second block (i.e., between time 1’00” and 1’30”) is generated, given the songs in our test domain tend to have introductory non-vocal part in the first 60 seconds. Then the content features of the 30 second block are extracted using the approach described in section 3.2. After feature extraction, each music track is represented as a 80-dimensional feature vector: $F_i = (F_{i,1},\cdots,F_{i,80})$. As described in Section 3.2, the first 12 features are based on the magnitude of the Short Time Fourier Transform (STFT) (e.g., means and variances of Spectral Centroid, Rolloff, Flux, Zero Crossings, and Low Energy), the next 52 features represents the means and variances of Mel-Frequency Cepstral Coefficients (MFCC), and the last 16 features are DWCH features.

User Access Pattern Representation

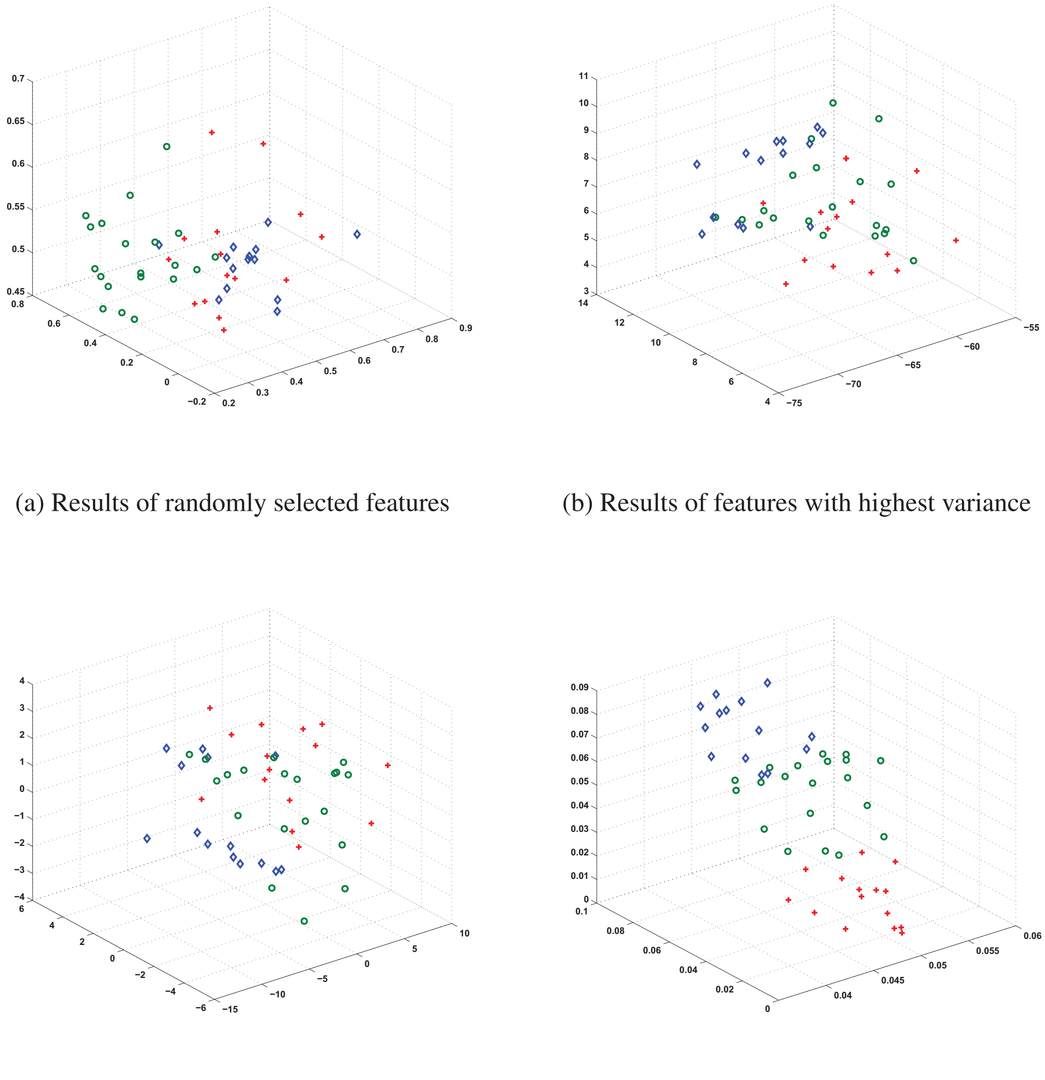
The access pattern of a user is represented as a 0/1-vector. Its dimension is equal to the number of songs available. For each *i*, the *i*-th entry of the vector is 1 if the user added the song in his/her playlist and 0 otherwise.

Recommendation List Generation

By combining the user access pattern data with the content features of the songs, the weight is generated for each feature using the dynamic weighting scheme described above. Then the music ranking algorithm aforementioned is employed to output the desired number of music pieces as our recommendations. In the experiments, the values of the ratings for the seed songs are set to be the same.

5.4.3 *Evaluation on Dynamic Weighting Schemes*

First of all, the performance of the dynamic weighting schemes is evaluated. In order to do so, we take a sample dataset consisting of 50 songs from three different classes. Note that the classes are determined by a group of users. Now we use the following methods to scatter positions of the 50 songs, and compare them in Figure 15. Note that each subfigure visualize the grouping results of different methods where each shape (there are three shapes: diamond, circle, and star) represents a class of songs.



(a) Results of randomly selected features

(b) Results of features with highest variance

(c) Results of PCA

(d) Results of weighted features

Figure 15: Evaluation on weighting schemes

1. Randomly select three original content features and scattering the position of each song based on these features.

2. Choose three content features with highest variances and scattering positions of the 50 songs.

3. Use principal components analysis (PCA) to select three principal components associated with the largest eigenvalues of the covariance matrix.

4. Choose three features with the highest weights by the dynamic weighting scheme (DWA).

From Figure 15, we observe that the Dynamic Weighting Approach (DWA) outperforms the other feature selection methods in separating three groups of songs: the features selected by DWA are highly relevant to the grouping. It shows that the features associated with the learned weighted from the user access patterns have the description power to distinguish the music pieces, while features with large variances or covariances do not help much in this case.

5.4.4 Comparison on Different Recommendation Approaches

To demonstrate the performance of DWA, we compare the performance of the following five approaches:

- **Content-based Approach(CBA)** This is solely based on acoustic content features extracted from the pieces of songs.

- **Artist-based Approach(ABA)** This is solely based on artist, namely, it recommends songs only from the same artist.

- **Access-pattern-based Approach(APA)** This is based on user access patterns. It selects the top songs with the highest co-occurrence frequency in the same playlists with the input song. This can also be thought as the item-based collaborative filtering method.

- **Hybrid Approach(HA)** This is the approach explained in section 5.1. It tries to integrate the collaborative filtering method and content-based method based on the algorithms described in [65].

- **DWA** This is based on our approach, which first utilizes user access patterns to dynamically learn weights for each content features and then perform label propagation and ranking for music recommendation.

An Illustrating Example

App-roach	Artist	Title	Genre
1	Tu Honggang	Singing with Wines	Rock
	Leehom Wang	Revolution	R & B
	Teresa Teng	I Only Care About You	Folk
	Faye Wong	Half Way	Pop
	Fish Leong	Shining Star	Rock
2	Jay Chou	Sunny Day	Blues
	Jay Chou	Thousand Miles Away	Pop
	Jay Chou	Sorry	R&B
	Jay Chou	Happier Than Before	R&B
	Jay Chou	Last Campaign	R&B
3	Jay Chou	Thousand Miles Away	Pop
	Jay Chou	Happier Than Before	R&B
	Hongmin You	Sand Rain	Pop
	Jay Chou	Cute Lady	R&B
	Jay Chou	Nunchucks	R&B
4	Jay Chou	Chrysanthemum Terrace	R&B
	Jay Chou	Romance Mobile	R&B
	Tu Honggang	Singing with Wines	Rock
	Rong Zhong	The Everest	Folk
	Jolin Tsai	Disappearing Castle	Pop
5	Jay Chou	Sorry	R&B
	Leehom Wang	Revolution	R&B
	Jolin Tsai	Spirit of Knight	R&B
	Leehom Wang	Bamboo	R&B
	Fish Leong	Silk Road of Love	Rock

Table 12: An illustrating example of different recommendation approaches

Table 12 shows an example of recommendation results by the five approaches that have just been described. In this example, the seed piece (which the user is currently listening to and from which the recommendation approaches are expected to produce a list of recommended songs) is “Love Before Christ”, an R&B song by a popular Chinese singer, Jay Chou.

From Table 12, we can see that if the recommendation only bases on content features, the results are somehow messy. And if we recommend only the songs from the same artist, the results do not “surprise” users at all because everybody knows other songs of the same artist might be in a similar flavor. What users expect is a novel and refreshing recommendation. We observe that our approach can provide some songs from different artists and with similar genre. Actually these songs do relate to the input song because some of them are from the same composers or lyricists, and these artists are of the same style as well.

Based on the the data we collect and process, we conduct several sets of experiments to compare the performance of the listed approaches. The first two comparisons are designed to test the recommendation novelty and the playlist generation experiment is to examine the recommendation prediction ability, while the user study conducted is to assess the overall recommendation performance from the viewpoints of the end users.

Artist Variety Comparison

In this experiment, we evaluate how artist variety is achieved in different approaches. Since artist-based approach consider songs from the same artists, we only have to compare approach CBA, APA, HA and DWA. For each of the 2829 songs, 10 songs are chosen for the recommendation output. We count the number of distinct artists that the 10 songs come from. From the statistical results listed in Table 13, we can see that content-based approach and our dynamic-weighting approach recommend songs with the richest artist variety, which is better than the hybrid approach and the access-pattern-based approach.

Approach	CBA	APA	HA	DWA
Average Number of Artists	8	5	7	8

Table 13: Results for artist variety comparison. The numbers are rounded to integers to be practically meaningful.

Content Variety Comparison

In this experiment, we evaluate if content variety as described in 5.1 are well balanced in different approaches.

First of all, we cluster the 2829 songs using K-means algorithm according to their content features, and then, we study how many clusters the 10 songs recommended by each approach belong to. Also, we calculate the average distance among the 10 recommended songs of each of the 2829 seed songs using their content features. The more the clusters and/or the larger the distances, the more diverse the 10 songs, i.e. the more opportunity to get novel recommendation results.

Approach	Mean of Average Distance	Mean of Average Number of Clusters
CBA	2.55	2
ABA	8.74	5
APA	10.01	6
HA	5.28	4
DWA	5.88	4

Table 14: Results for content variety comparison

From the experimental results listed in Table 14, we can clearly observe that content-based approach recommends songs with the highest content similarity, and the variety is very low. On the contrary, the access-pattern-based approach and the artist-based approach are diverse enough but lack of content similarity. Hybrid approach and our dynamic-weighting approach have comparable performance in well-balancing the content variety.

Playlist Generation Comparison

Since playlists are generally a good means to reflect the interests of users, by comparing how accurate we can generate the whole original playlists from part of songs in them using different methods, we can analyze the ability of the approaches to predict the interests and preferences of the users.

In this set of experiments, we randomly select 200 playlists from the dataset of 274 playlists, and run hybrid approach and our dynamic-weighting approach on the data for the two approaches to learn. Then we randomly select 5 songs from each of the rest 74 playlists, and generate 74 new playlists, each of which contains 50 distinct songs based on the ordered recommendation lists of the these 5 songs. Then we check how many of the songs in the rest of each original playlists (the number of songs available for checking varies from 5 to 15) match the songs in the new larger playlists. Figure 16 lists the boxplot results of the comparison among content-based approach, hybrid approach, and our dynamic-weighting approach.

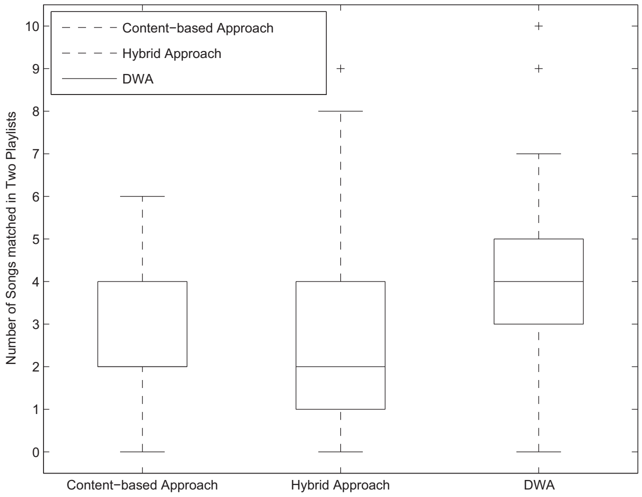


Figure 16: Number of songs matched in the original user playlists and the generated playlists

Approach	CBA	HA	DWA
Winning Rounds	8	20	37

Table 15: Times of one approach outperforms the other two approaches by comparing the matches in two playlists

From Figure 16 and Table 15, we clearly see that our DWA approach outperforms content-based approach and the hybrid approach. If we check the data in detail, we can find that for predicting some playlists, when there is enough song co-occurrence information, the hybrid approach works very well and have the comparable performance with our dynamic-approach. However, when dealing with new song sets and there are very little song co-occurrence data, the hybrid approach is almost degraded to content-based approach. On the contrary, our dynamic-weighting approach is trying to predict the recommended songs based on the weights already learned and the content features extracted, it can keep the similar performance when dealing with new song sets.

User Study

We develop a web interface and invite the users from the website to assess the recommendation results of different approaches. The interface can be found at:(`http://www.`

newwisdom.net/music/songUserStudy.jsp). For each song, we list the recommended songs (song titles and singers) using the five approaches described above. For each seed song that interests the user, he/she is invited to choose those that also interest him/her in the recommended list, and also select the best approach based on their perception. Note that the songs presented to the visitors are randomized and there is no fixed song appearance order. We asked the visitors to rate the recommended songs as well as the overall impression of all the five approaches for a given seed song.

To submit a feedback, the user must choose one and only one best approach from the five, but he/she can select any number of songs from the recommendation list as he/she likes. To make different songs have nearly equal chances to be exposed to the users for judgment, the selection of songs from the repository is also randomized. By collecting the IP addresses of the users, we know that more than 50 users (59 IP addresses) participated in the user study, and the recommendation results of 166 distinct songs are assessed by one or some of them. Altogether there are 201 submission of feedbacks. Table 16 lists the statistical results of the user study and Figure 17 compares the number of times people claim that an approach is the best among the five approaches.

	Approach				
	CBA	ABA	APA	HA	DWA
r_1	25	47	38	48	**69**
r_2	31	54	44	52	**60**
r_3	19	34	41	49	**52**
r_4	17	37	33	51	**58**
r_5	22	38	45	**47**	44
r_6	19	**49**	31	44	43
r_7	13	22	27	**52**	47
r_8	22	14	25	19	**42**
r_9	7	17	24	32	**39**
r_{10}	16	19	28	16	**38**
sum	191	331	336	410	**492**

Table 16: Results of the user study

The results of the user study were listed in Table 16. In the table, for each i, $1 \leq i \leq 10$, the row "r_i" shows the total number of times that songs at the ith position in the recommendation list is selected by users for each approach. The row "sum" lists the corresponding summation of all the values for each of the five approaches. By checking the statistical results of the user study, we can clearly see that our approach outperforms all the rest. For example, in row "r_1", there are 69 times that the recommended songs in position 1 by our dynamic-weighting approach are considered to be valuable recommendations while for hybrid method, there are only 48 times. In Figure 17, we also know that our dynamic-weighting approach is regarded as the best one among the five choices for most users at most times. Users sometimes also think the recommended songs from the same artists are what they prefer, but as we all know, that recommendation does not give users enough novel information.

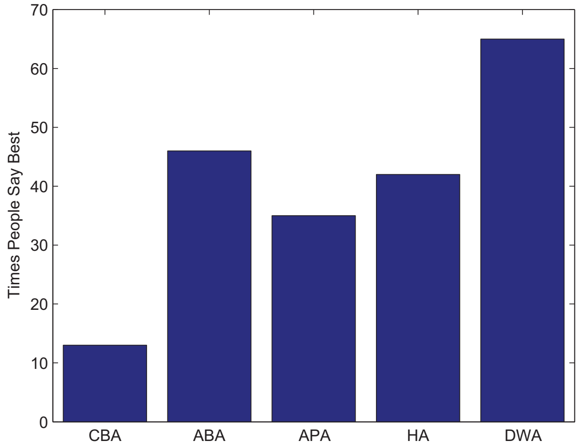


Figure 17: Times people say one approach is the best among all approaches

5.5 Conclusion

Both collaborative-filtering and content-based recommending schemes have their own advantages and limitations. In this paper, we propose a novel dynamic music similarity measurement scheme that integrates the acoustic content features and user access patterns. This scheme is based on the assumption that two pieces of music are similar in human perception when they share similar access patterns across multiple users. To calculate the new similarity measure, we use the metric learning approach, which learns appropriate similarity metrics based on the correlation between acoustic features and user access patterns of music, to automatically determine the weights for audio features. After obtaining the music similarity, music recommendation can be treated as a label propagation from labeled data (i.e., items with ratings) to unlabeled data. Comparing with other probabilistic models and hybrid approaches, our method incorporates the content similarity data and collaborative filtering information seamlessly. Experimental results and user study on a real data set demonstrate the recommendation quality of our proposed approach outperforms the others.

Although our proposed recommendation scheme has been tested to be effective, there are several venues for further research. One natural direction is to extent our current framework for personalized music recommendation. Furthermore, we can investigate more comprehensive music content features for similarity measurements.

CHAPTER 6

SYSTEM DEVELOPMENT

A prototype system for multi-modal music information retrieval and a real world user-centric music retrieval web application have been developed in the research study.

6.1 A Prototype System for Multi-modal Music Information Retrieval

This prototype system was the first attempt to evaluate our proposed techniques. The prototype system was implemented as a web application and it was able to: (a) provide a multi-model query interface for music information retrieval; (b) conduct genre classification off-line to help to build the system and offer the user a way to check the genre of the search music online; (c) summarize music pieces off-line and present the audio thumbnails to the users so that the results could be easily digested; (d) keep track of user listening behaviors; and (e) invite the users to actively provide feedbacks. This served as a framework for us to further investigate the key issues to improve music information retrieval.

Text-based search and content-based search were implemented in the system. Text-based search asks the user to input a piece of text information which can be song title, lyrics (sample piece), album title, artist, and/or genre. The returned results have similar titles, lyrics, etc.

Content-based search requires the user to provide a sample music piece. The system automatically extracts the content features from the music sample and compares the extracted features with the features of each song in our database. The feature extraction process was described in section 3.2. By default, the system returns the top 10 similar music pieces.

When users listen to (click) a song in the result set, their listening behaviors are recorded in the system. We also invite them to rate the search results.

6.1.1 System Architecture and Design

Figure 18 shows the typical 3-tier architecture we have adopted in this Web application. The Web Interface client uses the HTTP protocol to submit requests to the Query Processing

Engine, and this Engine receives the data from the Music Database using JDBC. The User Feedback Module aims to record the users' listening behaviors and obtain the active user feedbacks. The information is maintained and accumulated in the User Access Pattern Database. We use this information to adjust or tune the searches.

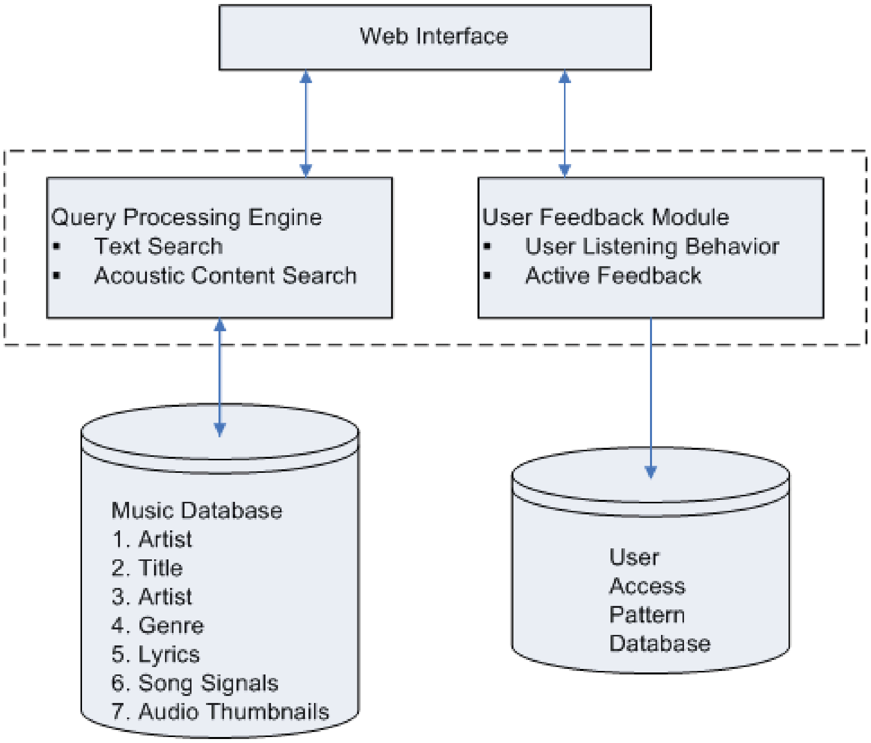


Figure 18: The system architecture

The summary of each module is listed as follows:

- Music Database: This module deals with the music data collection of each individual user. It enables browsing and sorting music pieces based on indexed keys. The database contains the information for about 800 songs including their signals, lyrics, artists, titles, composers etc.

- User Access Pattern Database: This module collects and stores access patterns for each individual user. The user access patterns can be used to identify user communities.

- Query Processing Engine: The module takes a user query as input, translates it into an executable form, executes the actual search, and presents the search results. The system supports text-based search and content-based search.

- User Feedback Module: This module collects the user feedback.
- Web Interface: This module provides a web-based interface for the users to communicate with the system.

6.1.2 Similarity Search

In text-based search, the matched results are returned in prioritized order. Exact match (e.g., match of words in the same order, match with full words, or match with words appearing in the title) has higher priority over the non-exact match (e.g., match of words in different orders, match with partial words, match with words appearing only in the lyrics).

In acoustic content-based search, we compare the content features extracted from the sample music with the features of the corresponding songs in database. The feature extraction process was described in section 3.2. After feature extraction, we represent each music track as a 80-dimensional feature vector: $S_i = (S_{i1}, \cdots, S_{i80})$. We normalize each dimension of the vector by subtracting the mean of that dimension across all the tracks and then dividing it by the standard deviation. The normalized representation vector is $\overline{S_i} = (\overline{S_{i1}}, \cdots, \overline{S_{i80}})$, where

$$\overline{S_{ij}} = \frac{S_{ij} - Mean(V_j)}{std(V_j)}, 1 \leqslant i \leqslant 80. \tag{27}$$

After normalization, we compute the Euclidean distance between the normalized representations. The 10 tracks with the shortest distances to the query are returned. If a user provides both types of the query, the system merges the top ranked results from text-based search and content- based search.

6.1.3 Genre Classification

To automatically get the genre of a song, we perform classification based on the acoustic features. Support Vector Machine classification algorithm (SMO from Weka) is employed for genre classification. The accuracy of the classification is about 85%. Since many songs do not have their genres when we built the system, we conduct the genre classification on those songs and complement the genre records for them.

6.1.4 Audio Summarization

The approach proposed in [30] was adopted to create audio thumbnails of all the music pieces in the music database. The basic idea is to find the segment with maximum similarity to the whole song based on self-similarity analysis. As the music summary is a continuous excerpt of the whole music piece, it sometimes cannot contain all segmentations such as introduction, verses, and refrains. But continuity does make the summary more natural when presented to the end users.

With the help of jAudio [95], this work was done offline and a 30-second music excerpt was generated for each song in the music database. MFCC feature set was used to represent the audio for similarity comparison as suggested in [30].

6.1.5 A Case Study

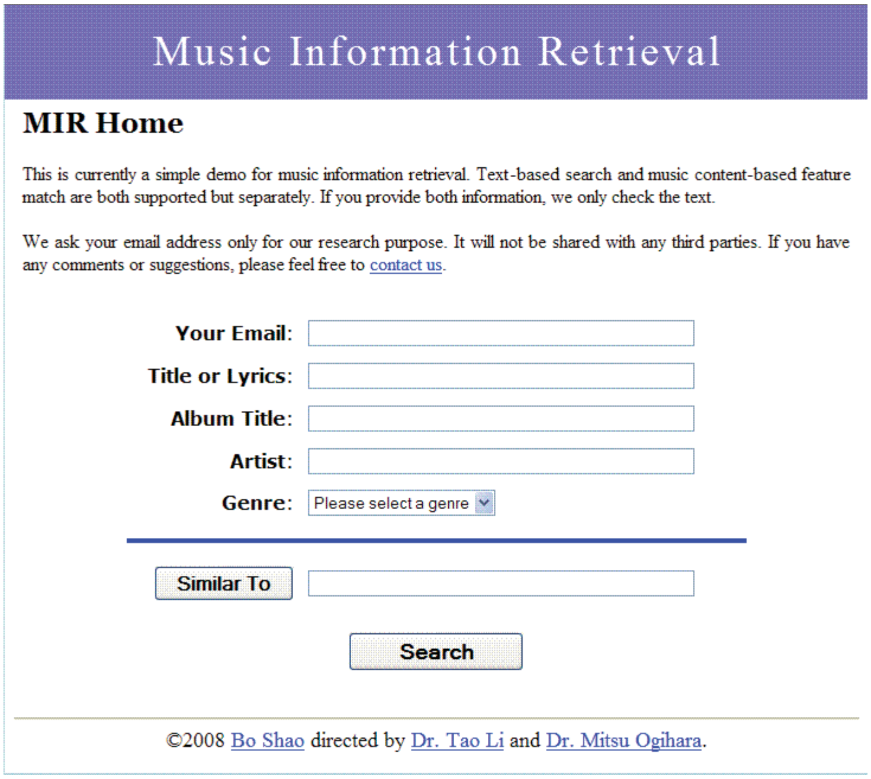
We list a few user interfaces to show some basic functions implemented in this framework.

Figure 19 shows the main user interface where visitors can input their search conditions including the text and acoustic content information and start search.

Figure 20 shows an example of searching based on the acoustic content information. The top 10 results are returned as a default. We invite the visitors to actively rate the search results to improve our search algorithm.

When the user tries to listen to a song by clicking the link, the system records the following information: 1) user ID information including the user's email, session ID, and IP address; 2) the song ID; 3) the time when the user clicks the link; and 4) search input including the concrete text information and the search type, i.e, search by text or by acoustic content.

The user access pattern information collected here will be used to improve our music similarity measures, facilitate personalized music recommendation, and so on.



Music Information Retrieval

MIR Home

This is currently a simple demo for music information retrieval. Text-based search and music content-based feature match are both supported but separately. If you provide both information, we only check the text.

We ask your email address only for our research purpose. It will not be shared with any third parties. If you have any comments or suggestions, please feel free to contact us.

©2008 Bo Shao directed by Dr. Tao Li and Dr. Mitsu Ogihara.

Figure 19: Main user interface

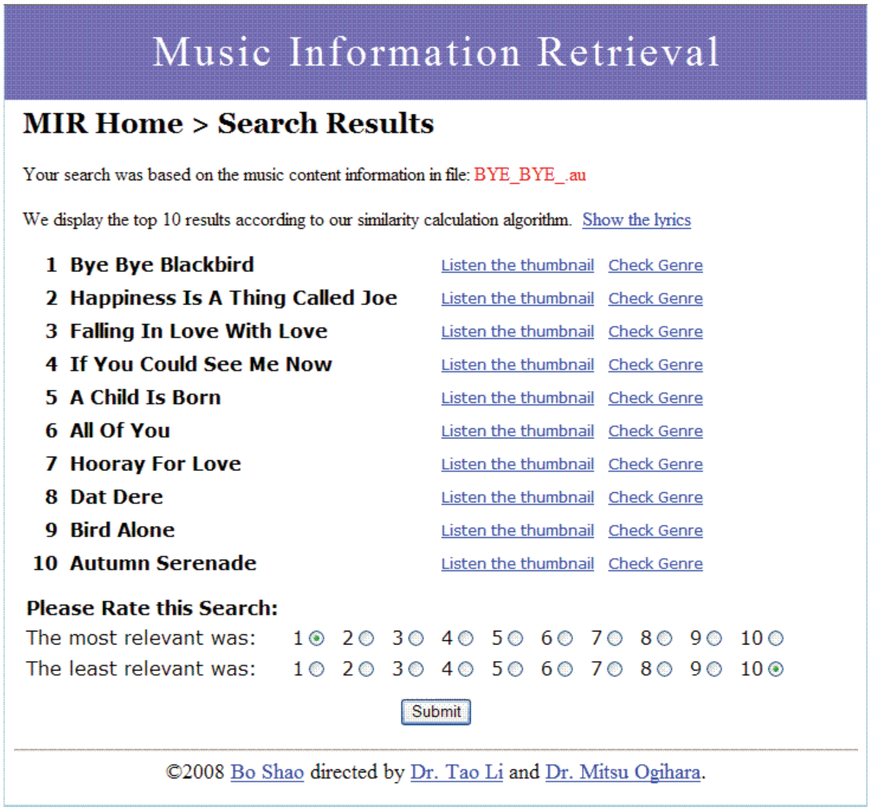
6.2 A Real World User-Centric Music Retrieval Web Application

Following this prototype system, a real world user-centric music retrieval web application is being developed. In order to get the user access pattern from real web visitors, this web application was embedded inside a public website `http://www.newwisdom.net`. This system includes mainly the end user interface, the back end data management interface, and a light weight web crawler to automatically collect music data. To avoid legal issues on music copyrights, this website is not a commercial website, and the music system we developed and the user access data the system collected are solely used for research purposes.

6.2.1 Main Functionalities Available to End Users

Here are the basic functionalities provided to the end users:

1. Visitors can search songs by singer name, singer popular aliases, album, and/or part of the lyrics. Visitors can also browse songs via other navigation approaches such as genre organization.



Music Information Retrieval

MIR Home > Search Results

Your search was based on the music content information in file: BYE_BYE_.au

We display the top 10 results according to our similarity calculation algorithm. Show the lyrics

1	Bye Bye Blackbird	Listen the thumbnail	Check Genre
2	Happiness Is A Thing Called Joe	Listen the thumbnail	Check Genre
3	Falling In Love With Love	Listen the thumbnail	Check Genre
4	If You Could See Me Now	Listen the thumbnail	Check Genre
5	A Child Is Born	Listen the thumbnail	Check Genre
6	All Of You	Listen the thumbnail	Check Genre
7	Hooray For Love	Listen the thumbnail	Check Genre
8	Dat Dere	Listen the thumbnail	Check Genre
9	Bird Alone	Listen the thumbnail	Check Genre
10	Autumn Serenade	Listen the thumbnail	Check Genre

Please Rate this Search:

The most relevant was: 1 ● 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○

The least relevant was: 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ●

©2008 Bo Shao directed by Dr. Tao Li and Dr. Mitsu Ogihara.

Figure 20: An example of the content-based search

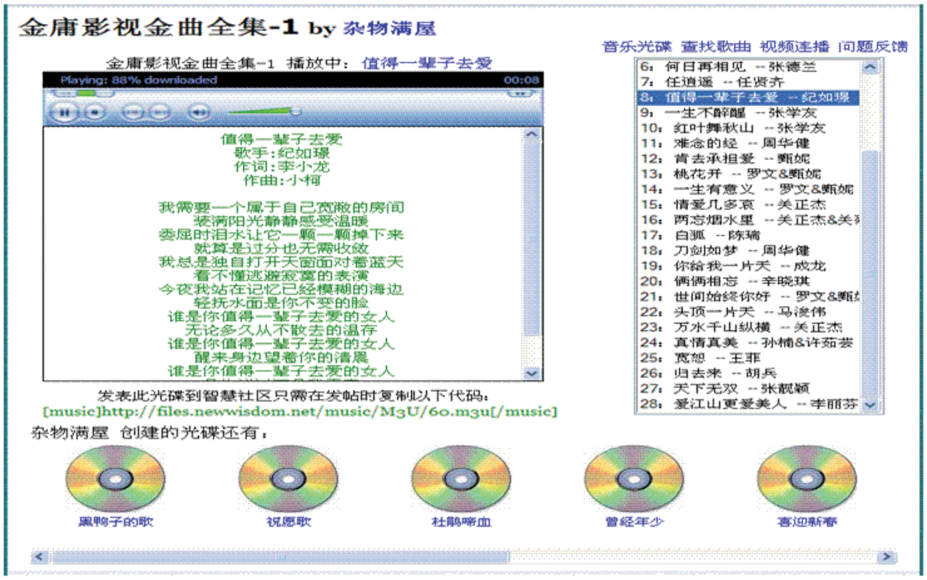
2. Registered users are encouraged to add/upload songs, provide metadata and lyrics information. The songs can be the favorite song not available yet in the system or their own creative music work. Visitors can also edit/update the information they provided. They can also save their favorite songs in their profiles.

3. Registered users can create and update playlists. They have easy access to the playlists they own. After the user creates an audio playlist, the system will automatically create a video playlist if there are any songs that have the corresponding video information from `http://www.youtube.com`.

4. Registered users can play songs, albums, and playlists created by any users.

5. Registered users can communicate to each other, or recommend songs and/or playlists via on-site messages.

Figure 21 shows a sample interface of a user playing a playlist created by another user. Other playlists created by the second user were recommended to the first user.



金庸影视金曲全集-1 by 杂物满屋

金庸影视金曲全集-1 播放中：值得一辈子去爱

Playing: 88% downloaded 00:08

值得一辈子去爱
歌手:纪如璟
作词:李小龙
作曲:小柯

我需要一个属于自己宽敞的房间
装满阳光静静感受温暖
委屈时泪水让它一颗一颗掉下来
就算是过分也无需收敛
我总是独自打开天窗面对着蓝天
看不懂逃避寂寞的表演
今夜我站在记忆已经模糊的海边
轻抚水面是你不变的脸
谁是你值得一辈子去爱的女人
无论多久从不散去的温存
谁是你值得一辈子去爱的女人
醒来身边望着你的清晨
谁是你值得一辈子去爱的女人

6：何日再相见 --张德兰
7：任逍遥 --任贤齐
8：值得一辈子去爱 --纪如璟
9：一生不醉醒 --张学友
10：红叶舞秋山 --张学友
11：难念的经 --周华健
12：肯去承担爱 --甄妮
13：桃花开 --罗文&甄妮
14：一生有意义 --罗文&甄妮
15：情爱几多哀 --关正杰
16：两忘烟水里 --关正杰&关
17：白狐 --陈瑞
18：刀剑如梦 --周华健
19：你给我一片天 --成龙
20：俩俩相忘 --辛晓琪
21：世间始终你好 --罗文&甄
22：头顶一片天 --马浚伟
23：万水千山纵横 --关正杰
24：真情真美 --孙楠&许茹芸
25：宽恕 --王菲
26：归去来 --胡兵
27：天下无双 --张靓颖
28：爱江山更爱美人 --李丽芬

发表此光碟到智慧社区只需在发帖时复制以下代码：
[music]http://files.newwisdom.net/music/M3U/60.m3u[/music]

杂物满屋 创建的光碟还有：

Figure 21: An example of a user playing a playlist

A new functionality of inviting user to provide review tags to singers, albums, playlists and songs, and search by tag combinations is under development. Any existing functionalities in the prototype system will be integrated into this real world application.

6.2.2 Music Data Management Interfaces

Here are the major functionalities for the data management available to the authorized users like the webmaster.

1. The authorized users have an overall view of all music data available in the system.

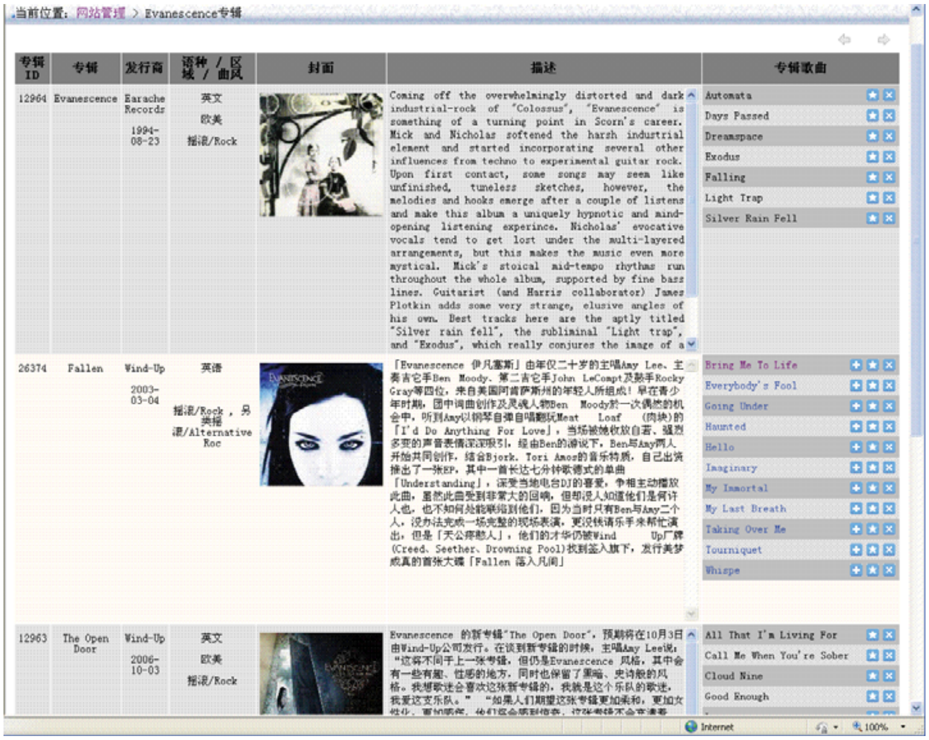
2. The authorized users can edit almost all metadata created by the visitors or the web crawler.

3. The authorized users can delete playlists.

4. The authorized users can block malicious visitors from accessing the system.

5. The authorized users can control publishing/unpublishing a music piece to public.

Figure 22 shows a sample interface of an authorized user viewing/editing the information of all albums from the singer: Evanescence. We can see that the album name, publisher, publishing date, main language, country/area, main genre, album cover image, description, and all songs in the albums are listed. With a double click on the description, the content of the description can be updated. When the actual music audio data is available for a song, a hyperlink is presented to the user so that he/she can listen to the song. The controls on the right side of a song are about to change the song title, delete the song from the album, and publish the song to the public.



当前位置：网站管理 > Evanescence专辑

专辑ID	专辑	发行商	语种/区域/曲风	封面	描述	专辑歌曲
12964	Evanescence	Earache Records 1994-08-23	英文 欧美 摇滚/Rock		Coming off the overwhelmingly distorted and dark industrial-rock of "Colossus", "Evanescence" is something of a turning point in Scorn's career. Mick and Nicholas softened the harsh industrial element and started incorporating several other influences from techno to experimental guitar rock. Upon first contact, some songs may seem like unfinished, tuneless sketches, however, the melodies and hooks emerge after a couple of listens and make this album a uniquely hypnotic and mind-opening listening experince. Nicholas' evocative vocals tend to get lost under the multi-layered arrangements, but this makes the music even more mystical. Mick's stoical mid-tempo rhythms run throughout the whole album, supported by fine bass lines. Guitarist (and Harris collaborator) James Plotkin adds some very strange, elusive angles of his own. Best tracks here are the aptly titled "Silver rain fell", the subliminal "Light trap", and "Exodus", which really conjures the image of a	Automata Days Passed Dreamspace Exodus Falling Light Trap Silver Rain Fell
26374	Fallen	Wind-Up 2003-03-04	英语 摇滚/Rock，另类摇滚/Alternative Roc		「Evanescence 伊凡塞斯」由年仅二十岁的主唱Amy Lee、主奏吉它手Ben Moody、第二吉它手John LeCompt及鼓手Rocky Gray等四位，来自美国阿肯萨斯州的年轻人所组成！早在青少年时期，团中词曲创作及灵魂人物Ben Moody於一次偶然的机会中，听到Amy以钢琴自弹自唱翻玩Meat Loaf (肉块)的「I'd Do Anything For Love」，当场被她收放自若、强烈多变的声音表情深深吸引，经由Ben的游说下，Ben与Amy两人开始共同创作，结合Bjork、Tori Amos的音乐特质，自己出资推出了一张EP，其中一首长达七分钟歌德式的单曲「Understanding」，深受当地电台DJ的喜爱，争相主动播放此曲，虽然此曲受到非常大的回响，但却没人知道他们是何许人也，也不知何处能联络到他们，因为当时只有Ben与Amy二个人，没办法完成一场完整的现场表演，更没钱请乐手来帮忙演出，但是「天公疼憨人」，他们的才华仍被Wind Up厂牌(Creed、Seether、Drowning Pool)找到签入旗下，发行美梦成真的首张大碟「Fallen 落入凡间」	Bring Me To Life Everybody's Fool Going Under Haunted Hello Imaginary My Immortal My Last Breath Taking Over Me Tourniquet Whispe
12963	The Open Door	Wind-Up 2006-10-03	英文 欧美 摇滚/Rock		Evanescence 的新专辑"The Open Door"，预期将在10月3日由Wind-Up公司发行。在谈到新专辑的时候，主唱Amy Lee说："这将不同于上一张专辑，但仍是Evanescence 风格，其中会有一些有趣、性感的地方，同时也保留了黑暗、史诗般的风格。我想歌迷会喜欢这张新专辑的，我就是这个乐队的歌迷，我爱这支乐队。" "如果人们期望这张专辑更加柔和，更加女	All That I'm Living For Call Me When You're Sober Cloud Nine Good Enough

Figure 22: An example of an authorized user viewing or editing the albums of a singer

6.2.3 A Light Weight Web Crawler

To conduct a research on music information retrieval, it is essential to have access to a large volume of music pieces covering many genres and styles. In order to perform hierarchical music classification, to implement personalized music recommendation, and to attract more visitors, we also need a considerable number of music samples and the corresponding metadata. Developing a web crawler was one of the big efforts to expand the music database.

For practical purpose, the web crawler was designed to collect music data from a limited set of music websites. Configuration was provided for most part of it in order to adapt certain changes of those websites. In the case that certain web contents were generated by JavaScript, the crawler can simulate the functionality of a stand web browser and generate such contents. Exceptions such as server or network error can be recorded to initiate a future retry of the same page. Currently, it can navigate among the web pages and identify music metadata including information of singers, albums, and songs:

- Information of singers: Including singer name/alias, gender or type (singer or band), main language, country or area, singer portrait and introduction, and certainly the albums created by the singers.
- Information of albums: Including album name, main language, publisher, publish date, main genre, album cover image and description, and most importantly the songs in the albums.
- Information of songs: Including song title, music genre, lyrics, music audio, the link of corresponding video from `http://www.youtube.com`, and for sure the singer and album information of the songs.

With the work of the web crawler and the help of web visitors, now this music system has collected information about 13,000 singers with 44,000 albums from them, and metadata of 480,000 songs with about 300,000 unique music pieces, about 60% of which have corresponding video information. As the bandwidth of the website is very limited due to budget, only about 10-20% of the information can be published and really accessible to the web visitors. But it still has about 12,000 registered visitors and about 1,000 playlists created by them.

The music recommendation research work presented in chapter 5 and some of the work presented in chapter 3 were based on the music audio data and user access pattern information gathered from this system.

6.2.4 User Access Pattern and Personalized Music Recommendation

The playlists the users created have been a reliable resource of user access patterns. Our system has also been keeping track of the users' detailed access history, including the details of each listening activity (such as date, time, music title, artist, album, genre, and duration). Specifically, in order to record more accurate listening patterns, the system does not treat the action of clicking a song hyperlink as a listening activity, but waits until the user actually finishes listening the music piece. This was achieved via an AJAX request sent to the server when the browser-embedded player reaches its very end of playing the song.

In the previous work presented in chapter 5, we have established a dynamic music similarity measurement strategy. This is being actively integrated into the system. The basic idea is to first cluster the registered users based on their listening history into multiple groups, then apply the dynamic music similarity measurement strategy we proposed to generate the recommended music items for each group of users. This is under development. With this implemented, users in different groups will get different recommendations even given with the same seed song.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summary of Major Research Work and Contribution

In the effort of developing a user-centric music information retrieval system, we have performed numerous research studies. We have learned the existing algorithms in the literature and investigated multiple approaches that can be utilized in the application of music information retrieval. We developed serval useful algorithms and successfully applied them in our research. We also conducted significant amount of experiments, including necessary user studies, to evaluate the proposed algorithms and approaches. Finally we developed a prototype system and a real world application to assist our research work. The major work and contribution of the dissertation can be summarized as follows:

1. Developed a novel dynamic music similarity measurement strategy based on the proposed *dynamic weighting scheme* by incorporating collaborative-filtering approach and content-based approach. The dynamic weighting scheme is based on the assumption that two pieces of music are similar in human perception when they share similar access patterns across multiple users. To calculate the new similarity measure, we use the metric learning approach, which learns appropriate similarity metrics based on the correlation between acoustic features and user access patterns of music, to automatically determine the weights for audio features. After obtaining the music similarity, we treated music recommendation as a label propagation over a song graph from labeled data to unlabeled data. This approach seamlessly integrates the acoustic content and user access pattern data. The performance of this approach will not degrade when processing the audio content data that does not have corresponding user access pattern information. This approach has been successfully applied to the system that we developed to perform music recommendation. It has been tested with multiple experiments including user

studies, and the performance has been proved to be better than many other existing music recommendation approaches.

2. Developed a new approach, namely *hierarchical co-clustering algorithm*, to quantify the music artist similarity by employing the artist style and mood information extracted from *All Music Guide*. This algorithm is able to represent the style/mood term similarity by creating taxonomies. The term similarities are then quantified by capturing the positions of the terms in the generated taxonomies. Artist similarity is then calculated based on the style/mood term similarities. The quantified artist similarity has been validated by the acoustic features extracted from the music pieces produced by the artists. This effort facilitates the music artist organization and annotation in the overall music information retrieval task.

3. Proposed a multi-label classification approach, called *Hypergraph integrated Support Vector Machine (HiSVM)*, which can integrate several types of music information including music audio features, music style correlations, and social tag information and correlations. This enables the classifier to assign multiple styles to music objects in the classification.

4. Addressed the issue of clustering pop music pieces into groups with respect to the artists from diverse information sources. In order to effectively analyze music utilizing information from multiple modal data, we developed *bimodal music clustering algorithm* for integrating the features based on minimizing disagreement between different data sources. This algorithm can be considered as a kind of semantic integration of data from multiple sources, and it can implicitly learn the correlation structure between different sets of features. We also developed a *music constraint-based clustering framework* for clustering music songs in the presence of constraints.

5. A prototype system for multi-modal music information retrieval was developed and the combined search based on text as well as music content were implemented. The

techniques of music genre classification and music audio summarization have been studies from the literature and applied to the prototype system.

6. A real world user-centric music retrieval web application has been developed. User retrieval interface and music data management module were implemented. Specifically, a light weight web crawler was designed and implemented to expand the music database. Playlists, as very reliable and informative user access pattern data, are created by the registered users of the website on which the web application is hosted. Other user listening activities are also recorded and have been employed for personalized music recommendation functionality module that is under active development. This system has greatly assisted the past research activities and will continue to help the future research studies.

7.2 Potential Applications in Other Fields

Several algorithms, frameworks and approaches have been developed in this dissertation. Experiments have been conducted based on music data to validate these developed algorithms, and necessary user studies have been performed to demonstrate the effectiveness of these approaches in music information retrieval research area. One might naturally pose a question: can these algorithms, frameworks and approaches be applied to handle other types of data and be utilized to address the problems in other research areas? Our answer is yes. Let us take a deeper look at a few algorithms and approaches and explain which areas they can be applied to, and how they can be utilized.

The first example is the dynamic weighting scheme. This scheme has been successfully employed to learn the audio feature weights to measure audio similarity based on the user access pattern. It can be easily applied to video recommendation tasks. If we can extract video features and obtain corresponding user access pattern, we can use the same strategy to measure the video similarity and conduct video recommendation.

The second example is the hierarchical co-clustering algorithm and the artist similarity qualification framework. This algorithm and framework can be applied to multiple research areas, such as the quantification of document similarity. We can extract representative keywords from each document, and using hierarchical co-clustering algorithms to generate term taxonomies to quantify the term similarity, and finally we can calculate the document similarity. This idea can also be utilized to search similar documents on the web. Another useful area might be to look for similar items on a web store if the items have user-assigned tags. We can firstly quantify the similarity of the tags using the algorithm, and then quality the similarity among the items from the web store.

From these examples, we can understand that the developed approaches can be adopted or adapted to other research areas although they were only applied to music information retrieval tasks in this dissertation.

7.3 Future Work

As the user-centric music information retrieval system is such a complicated system, we have been making constant efforts to improve it. In the near future, we plan to perform the following studies:

7.3.1 Automatic Music Genre Classification in Large Taxonomies

In the current system, we have done the basic music genre classification. As we are expanding our music database, we will include genre classes, and these classes will be organized in a hierarchical tree such that related music classes are linked to the same nodes. To reach this goal, we will firstly classify the music pieces into one of the internal nodes in the hierarchy, then classify it into one of the music classes under the internal node. Each step of classification will only involve a very limited number of classes, thus the classification is more manageable and efficient than the direct approach. We plan to investigate the Hierarchical Mixture of Expert (HME) model [64] for the classification.

One challenge of genre classification is that the genres are not always mutually exclusive; that is, some music may be classified into more than one (but not many) genre. The problem can be considered as a multi-label classification [133] and corresponding algorithms will be designed to address it.

7.3.2 Relevance Feedback for Music Retrieval

When the initial retrieval results are unsatisfactory, relevance feedback methods [154] will be applied to improve the quality of retrieval results. We consider two different scenarios for relevance feedback:

1. Relative relevance judgements. Most relevance feedback techniques assume that users are able to provide absolute judgement regarding the relevance of retrieved items. However, due to the complexity of music and that of human relevance judgement, users may fail to provide absolute relevance judgements instantly. Therefore, we will ask them to provide relative judgements instead, such as to rank the retrieved music according to its relevance to their interests. This can be done in many iterations so that the query will be refined step by step.

2. Explore collaborative access patterns. As aforementioned, in addition to the audio features, each piece of music is also represented by its access patterns by large numbers of users. Given that each song has two types of representations, we plan to investigate methods that are able to explore the correlation between the two representations to better utilize the relevance judgements.

7.3.3 Lyrics Summarization

Lyrics summarization is a very helpful way to reinforce automatic music summarization. We plan to adopt the machine learning approach proposed in [73] to address this problem. In particular, we will represent each sentence in a lyric using the following features:

1. Sentence length cut-off feature. Based on the assumption that short sentences tend not to be included in summaries, we use this binary feature to determine whether the length of a sentence exceeds a certain threshold so that this sentence should be kept.

2. Term frequency of thematic words. Thematic words in a lyric refer to the words that appear across multiple different sentences. By comparing the term frequency of thematic

words for a sentence with that for the whole lyric, we are able to measure the correlation of the sentence to the lyric.

3. Number of repeats. This feature indicates how many times a sentence is repeated within a lyric. Very often, we find a sentence to be a good summary when it is repeated multiple times in the lyric.

We will train a classifier that learns weights for the above features from training examples. With the estimated weights, each sentence will be scored based on the weighted sum of features, and the sentence with the highest score will be selected as the summary.

7.3.4 Clustered Presentation of Retrieved Music

Upon receiving a query from the user the system searches for music pieces that match the query. Those pieces whose estimated relevance reaches a certain threshold are presented to the user. A rank list is typically used. When the set of returned records is too large, it is very tedious and time-consuming for the user to try each music piece to find what they really want. To facilitate users to browse through a dauntingly long list, we will design a clustered view to present the large number of matches and a rank list augmented with summaries for presenting a small number of them.

1. A rank list presentation with audio summaries will be presented to the end user. The list will be divided into pages and shown vertically. Each entry will be presented with its score, disco-graphic information, labels, and lyrics summary if available. Also, audio summaries will be given and they will be concatenated into a single audio file. The page will be presented with a side-bar with a marker. The marker position will indicate which position in which summary is being played. The user then will be able to quickly go through the whole list.

2. When there are many matches, the rank list approach in the above will not be effective. So we propose to develop a two dimensional cluster presentation. We envision a plot in a table, the row corresponding to genre and the column to mood. The retrieved data will

be assigned to the locations in the table that match their metadata information. Then the data will be hierarchically clustered row-wise and column-wise.

LIST OF REFERENCES

[1] S. Abeny. Bootstrapping. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367. Morgan Kaufmann Publishers, 2002.

[2] M. Alghoniemy and A. H. Tewfik. User-defined music sequence retrieval. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 356–358, 2000.

[3] A. Anglade, Q. Mary, R. Ramirez, and S. Dixon. Genre classification using harmony rules induced from automatic chord transcriptions. In *Proceedings of the International Conference on Music Information Retrieval*, pages 669–674, 2009.

[4] J. J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proceedings of the International Conference on Music Information Retrieval*, pages 157–163, 2002.

[5] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[6] E. Batlle, J. Masip, and E. Guaus. Automatic song identification in noisy broadcast audio. In *Signal and Image Processing*, 2002.

[7] A. Berenzweig, A., Ellis, D. P. W., and S. Lawrence. Using voice segments to improve artist classification of music. In *AES 22nd International Conference*, pages 79–86, 2002.

[8] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[9] C. Berge. Graphs and hypergraphs. 1973.

[10] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[11] P. Berkhin. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71, 2006.

[12] E. Bill. Some advances in transformation-based parts of speech tagging. In *Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 722–727. American Association for Artificial Intelligence, 1994.

[13] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT'98)*, pages 92–100. ACM Press, 1998.

[14] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann, 1998.

[15] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000.

[16] J. J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *Proc. Of the 6th Int. Conf. on Digital Audio Effects (DAFx)*, 2003.

[17] R. Cai, C. Zhang, L. Zhang, and W. Y. Ma. Scalable music recommendation by search. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 1065–1074, 2007.

[18] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In *Workshop on Multimedia Signal Processing*, pages 169–173, 2002.

[19] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied. Robust sound modeling for song detection in broadcast audio. In *Proc. AES 112th Int. Conv*, pages 1–7, 2002.

[20] N. Casagrande, D. Eck, and B. K. Geometry in sound: a speech/music audio classifier inspired by an image classifier. In *Proc. Of the Int. Computer Music Conferecnce (ICMC)*, 2005.

[21] W. Chai and B. Vercoe. Music thumbnailing via structural analysis. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 223 – 226, 2003.

[22] W. Chai and B. Vercoe. Structural analysis of musical signals for indexing and thumbnailing. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 27 – 34, 2003.

[23] C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines, 2001.

[24] H. C. Chen and A. L. P. Chen. A music recommendation system based on music data grouping and user interests. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 231–238, New York, NY, USA, 2001. ACM.

[25] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *Proceedings of The 4th SIAM Data Mining Conference*, pages 22–24, April 2004.

[26] F. R. K. Chung. The laplacian of a hypergraph. *Expanding Graphs, DIMACS Series*, 1993.

[27] R. Cilibrasi, P. Vitányi, and R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2004.

[28] W. W. Cohen and W. Fan. Web-collaborative filtering: recommending music by crawling the web. *Comput. Networks*, 33(1-6):685–698, 2000.

[29] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[30] M. Cooper and J. Foote. Automatic music summarization via similarity analysis. In *Proceedings of 3rd International Symposium on Music Information Retrieval*, pages 81–85, 2002.

[31] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop*, pages 127–130, 2003.

[32] R. B. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.

[33] S. Dasgupta, M. L. Littman, and D. McAllester. PAC generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 375–382, Cambridge, MA, 2002. The MIT Press.

[34] I. Davidson and S. S. Ravi. Clustering with constraints: feasibility issues and the k-means algorithm. In *Proceedings of the SIAM International Conference on Data Mining*, 2005.

[35] Virginia R. De Sa and Dana Ballard. Category learning through multi-modality sensing. *Neural Computation*, 10(5):1097–1117, 1998.

[36] C. DeCoro, Z. Barutcuoglu, and R. Fiebrink. Bayesian aggregation for hierarchical genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 77–80, 2007.

[37] O. Dekel, C. D. Manning, and Y. Singer. Log-linear models for label ranking, 2003.

[38] F. Deliège, B.Y. Chua, and T.B. Pedersen. High-Level Audio Features: Distributed Extraction and Similarity Search. pages 565–570, 2008.

[39] 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*, 39(1):1, 38 1977.

[40] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. Technical report, Department of Computer Science, University of Texas at Austin, 2001.

[41] C. Ding, R. Jin, T. Li, and H. D. Simon. A learning framework using green's function and kernel regularization with application to recommender system. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 260–269, New York, NY, USA, 2007. ACM.

[42] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135, 2006.

[43] W. J. Dowling and D. L. Harwood. *Music Cognition*. Academic Press, Inc, 1986.

[44] J. S. Downie. Toward the scientific evaluation of music information retrieval systems. In *Proceedings of 4th International Symposium on Music Information Retrieval*, pages 25–32, 2003.

[45] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proceedings of NIPS*, 2001.

[46] D. P. W. Ellis and G. E. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proceedings of ICASSP*, 2007.

[47] D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of 3rd International Conference on Music Information Retrieval*, pages 170–177, 2002.

[48] J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *Proceedings of the International Conference on Music Information Retrieval*, pages 265–266, 2002.

[49] J. Foote and S. Uchihashi. The beat spectrum: a new approach to rhythm analysis. In *IEEE International Conference on Multimedia & Expo 2001*, 2001.

[50] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H.G. Okuno. Singer identification based on accompaniment sound reduction and reliable frame selection. In *Proceedings of the International Conference on Music Information Retrieval*, pages 329–336, 2005.

[51] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London and New York, 1981.

[52] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *In Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 341–352, 2005.

[53] J. Haitsma, T. Kalker, and J. Oostveen. Robust audio hashing for content identification. In *Proc. of the Content-Based Multimedia Indexing*, 2001.

[54] J. A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[55] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold ranking based image retrieval. In *Proceedings of ACM Multimedia 2004*, 2004.

[56] X. He, W. Y. Ma, and H. J. Zhang. Learning an image manifold for retrieval. In *Proceedings of ACM MM 2004*, 2004.

[57] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, 1999.

[58] J.L. Hsu, C.C. Liu, and L.P. Chen. Discovering nontrivial repeating patterns in music data. *IEEE Transactions on Multimedia*, 3:311–325, 2001.

[59] X. Hu, J. S. Downie, K. West, and A. Ehmann. Mining music reviews: promising preliminary results. In *Proceedings of the International Conference on Music Information Retrieval*, pages 536–539, 2005.

[60] Y. C. Huang and S. K. Jenor. An audio recommendation system based on audio signature description scheme in mpeg-7 audio. In *2004 IEEE International Conference on Multimedia and Expo*, volume 1, pages 639–642, 2004.

[61] D. Huron. Perceptual and cognitive applications in music information retrieval. In *Proceedings of International Symposium on Music Information Retrieval*, 2000.

[62] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[63] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy, 1997.

[64] M. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[65] K. Y. Jung, D. H. Park, and J. H. Lee. Hybrid collaborative filtering and content-based filtering for improved recommender system. In *Computational Science - ICCS 2004*, pages 295–302. Springer Berlin / Heidelberg, 2004.

[66] S. Kim and S. Narayanan. Dynamic chroma feature vectors with applications to cover song identification. In *Proceedings of the International Workshop on Multimedia Signal Processing (MMSP)*, pages 984 – 987, 2008.

[67] Y. E. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *Proceedings of the International Conference on Music Information Retrieval*, pages 13–17, 2002.

[68] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pages 287–292, 2008.

[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, and G. Widmer. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 147–154, New York, NY, USA, 2006. ACM.

[71] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 2002 International Conference on Machine Learning (ICML)*, 2002.

[72] R. Kraft, Q. Lu, and S. Teng. Method and apparatus for music summarization and creation of audio summaries, 2001. US Patent 6,225,546.

[73] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of Research and Development in Information Retrieval*, pages 68–73, 1995.

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

[75] A.S. Lampropoulos, P.S. Lampropoulou, and G.A. Tsihrintzis. Musical genre classification enhanced by improved source separation techniques. In *Proceedings of the International Conference on Music Information Retrieval*, pages 576–581, 2005.

[76] J. H. Lee and J. S. Downie. Survey of music information needs, uses and seeking behaviours: Preliminary findings. In *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004.

[77] M. Li and R. Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the International Conference on Music Information Retrieval*, pages 252–259, 2005.

[78] Q. Li, B. M. Kim, D. H. Guan, and D. W. Oh. A music recommender based on audio features. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 532–533, New York, NY, USA, 2004. ACM.

[79] T. Li and L. Li. *Music Data Mining: An Introduction*. Chapman & Hall/CRC Press, 2011.

[80] T. Li and M. Ogihara. Content-based music similarity search and emotion detection. In *Proceedings of 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 705–708, 2004.

[81] T. Li and M. Ogihara. Music artist style identification by semisupervised learning from both lyrics and content. In *Proceedings of the ACM Conference on Multimeda*, pages 364–367, 2004.

[82] T. Li and M. Ogihara. Music genre classification with taxonomy. In *Proceedings of The 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, pages V198–201, 2005.

[83] T. Li and M. Ogihara. Semi-supervised learning from different information sources. *Knowledge and Information Systems Journal*, 7(3):289–309, 2005.

[84] T. Li and M. Ogihara. Towards intelligent music retrieval. *IEEE Transactions on Multimedia*, 2005.

[85] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proceedings of SIGIR*, pages 282–289. acm, 2003.

[86] T. Li, M. Ogihara, W. Peng, B. Shao, and S. Zhu. Music clustering with features from different information sources. *Trans. Multi.*, 11:477–485, April 2009.

[87] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[88] C. C. Liu and C. S. Huang. A singer identification technique for content-based classification of MP3 music objects. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 438–445. ACM, 2002.

[89] B. Logan. Music recommendation from song sets. In *Proceedings of ISMIR 2004*, pages 425–428, Oct 2004.

[90] B. Logan and S. Chu. Music summarization using key phrases. In *IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP00)*, volume 2, pages 749–752, 2000.

[91] B. Logan and A. Salomon. A content-based music similarity function. Technical Report CRL 2001/02, Cambrige Research Laboratory, jun 2001.

[92] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs, 2006.

[93] H. Lukashevich, J. Abeßer, C. Dittmar, and H. Grossmann. From multi-labeling to multi-domain-labeling: a novel two-dimensional approach to music genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 459–464, 2009.

[94] J. P. G. Mahedero, A. Martinez, P. Cano, M. Koppenberger, and F. Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th ACM International Conference on Multimedia (MULTIMEDIA 05)*, page 475C478, 2005.

[95] D. Mcennis, C. Mckay, I. Fujinaga, and P. Depalle. Jaudio: A feature extraction library. In *International Conference on Music Information Retrieval*, 2005.

[96] C. McKay. Automatic genre classification of midi recordings, 2004.

[97] C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of the International Conference on Music Information Retrieval*, pages 525–530, 2004.

[98] M. F. McKinney and J. Breebaart. Features for audio and music classification. In *Proceedings of ISMIR*, 2003.

[99] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 2002.

[100] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *Proceedings of the International Conference on Music Information Retrieval*, pages 604–609, 2005.

[101] R. Mitton. Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information Processing and Management*, 23(5):103–209, 1987.

[102] S. Monti, P. Tamayo, J. Mesirov, and T. Gloub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning Journal*, 52(1-2):91–118, 2003.

[103] M. Mueller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of ISMIR-05*, pages 288 – 295, 2005.

[104] H. Muller, T. Pun, and D. Squire. Learning from user behavior in image retrieval: Application of market basket analysis. *Int. J. Comput. Vision*, 56(1-2):65–77, 2004.

[105] R. Neumayer and A. Rauber. Multi-modal music information retrieval - visualisation and evaluation of clusterings by both audio and lyrics. In *Proceedings of the 8th RIAO Conference*, 2007.

[106] N. Oliver and L. Kreger-Stickles. Papa: Physiology and purpose-aware automatic playlist generation. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 250–253, October 2006.

[107] N. Orio. *Music retrieval: A tutorial and review*. 2006.

[108] F. Pachet and D. Cazaly. A taxonomy of musical genres. In *Proceedings of Content-Based Multimedia Information Access (RIAO)*, 2000.

[109] F. Pachet, P. Roy, and D. Cazaly. A combinatorial approach to content-based music selection. In *IEEE Multimedia*, pages 457–462, 2000.

[110] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 628–633, 2005.

[111] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proceedings of the ACM Multimedia 2002 (ACM MM2002)*, pages 570–579, 2002.

[112] C. Papaodysseus, G. Roussopoulos, D. Fragoulis, T. Panagopoulos, and C. Alexiou. A new approach to the automatic recognition of musical recordings. *J. Audio Eng. Soc.*, 49:23–35, 2001.

[113] S. Pauws, W. Verhaegh, and M. Vossen. Fast generation of optimal music playlists using local search. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 138–143, October 2006.

[114] G. Peeters, A. L. Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Proceedings of the International Conference on Music Information Retrieval*, 2002.

[115] W. Peng, T. Li, and M. Ogihara. Music clustering with constraints. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 27–32, 2007.

[116] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *Advances in Neural Information Processing Systems 14*, pages 1425–1432, 2002.

[117] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *17th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, Seattle, Washington, August 2–5 2001.

[118] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. 1993.

[119] R. Ragno, C. J. C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 73–80, New York, NY, USA, 2005. ACM.

[120] A. Rauber, E. Pampalk, and D. Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarities. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR)*, pages 71–79, 2002.

[121] J. Reed and C. H. Lee. A study on music genre classification based on universal acoustic models. In *Proceedings of the International Conference on Music Information Retrieval*, pages 89–94, 2006.

[122] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[123] D. Roth and D. Zelenko. Toward a theory of learning coherent concepts. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence (AAAI/IAAAI'00)*, pages 639–644. AAAI Press / The MIT Press, 2000.

[124] Y. Rui and T. S. Huang. Optimizing learning in image retrieval. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 236–243, 2000.

[125] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. In *ACM WebKDD Workshop*, 2000.

[126] N. Scaringella. Timbre and rhythmic trap-tandem features for music information retrieval. In *Proceedings of ISMIR*, 2008.

[127] N. Scaringella and G. Zoia. On the modeling of time information for automatic genre recognition systems in audio signals. In *In Proc. International Symposium on Music Information Retrieval*, pages 666–671, 2005.

[128] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: A survey. *IEEE Signal Processing Magazine*, 23:133–141, 2006.

[129] J. Ben Schafer, Joseph Konstan, and John Riedi. Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[130] M. Schedl, P. Knees, and G. Widmer. Discovering and visualizing prototypical artists by web-based cooccurrence analysis. In *Proceedings of ISMIR*, pages 21 – 28, 2005.

[131] J. R. A. Schlicker, F. S. Domingues, and T. Lengauer. A new measure for functional similarity of gene products based on gene ontology, 2006.

[132] D. Schnitzer, A. Flexer, G. Widmer, and A. Linz. A filter-and-refine indexing method for fast similarity search in millions of music tracks. pages 537–542, 2009.

[133] B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[134] B. Shao, T. Li, and M. Ogihara. Quantify music artist similarity based on style and mood. In *Proceeding of the 10th ACM workshop on Web information and data management*, WIDM '08, pages 119–124, 2008.

[135] B. Shao, D. Wang, T. Li, and M. Ogihara. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech and Language Processing*, 17:1602–1611, November 2009.

[136] J. Shen, B. Cui, J. Shepherd, and K. L. Tan. Towards efficient automated singer identification in large music databases. In *Proceedings of the 29th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 59–66. ACM, 2006.

[137] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *Proceedings of the International Conference on Music Information Retrieval*, pages 313–381, 2008.

[138] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, pages 144–158, 2003.

[139] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, March 2003.

[140] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multilabel classification. In *Proceedings of KDD*, 2008.

[141] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

[142] P. N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[143] W. H. Tsai, D. Rodgers, and H. M. Wang. Blind clustering of popular music recordings using singer voice characteristics. *Computer Music Journal*, 30(3):68–78, 2004.

[144] W. H. Tsai, H. M. Wang, and D. Rodgers. Automatic singer identification of popular music recordings via estimation and modeling of solo vocal signal. In *Eighth European Conference on Speech Communication and Technology*, pages 2993–2996, 2003.

[145] Y. Tsuchihashi, T. Kitahara, and H. Katayose. Using bass-line features for content-based mir. In *Proceedings of ISMIR*, 2008.

[146] G. Tzanetakis and P. Cook. Music genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[147] A. Uitdenbogerd and R. V. Schyndel. A review of factors affecting music recommender success. In *Proceedings of ISMIR 2002*, 2002.

[148] V. Vapnik. The nature of statistical learning theory, 1995.

[149] V. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[150] F. Wang, X. Wang, B. Shao, T. Li, and M. Ogihara. Tag integrated multi-Label music style classification with hypergraph. In *Proceedings of the International Conference on Music Information Retrieval*, pages 363–368, 2009.

[151] B. Wei, C. Zhang, and M. Ogihara. Keyword generation for lyrics. In *Proceedings of the Eighth International Conference on Music Information Retrieval (ISMIR07)*, pages 121–122, 2007.

[152] D. Wettschereck and D. W. Aha. Weighting features. In M. Veloso and A. Aamodt, editors, *Case-Based Reasoning, Research and Development, First International Conference*, pages 347–358. Springer-Verlag, Berlin, 1995.

[153] B. Whitman and D. Ellis. Automatic record reviews. In *Proceedings of ISMIR*, 2004.

[154] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the 2002 International Computer Music Conference*, pages 591–598, 2002.

[155] L. Wu, S. L. Oviatt, and P. R. Cohen. Multimodal integration - a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341, 1999.

[156] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512, 2003.

[157] C. Xu, N.C. Maddage, and X. Shao. Automatic music classification and summarization. *IEEE Transactions on Speech and Audio Processing*, 13(3):441–450, 2005.

[158] G. Xu. Building implicit links from content for forum search. In *SIGIR 06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 300–307. ACM Press, 2006.

[159] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proceedings of ISMIR*, 2006.

[160] H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Bipartite graph partitioning and data clustering. In *Proceedings of ACM CIKM (CIKM'01)*, 2001.

[161] T. Zhang. Automatic singer identification. In *Proceedings of the International Conference on Multimedia and Expo*, pages 33–36, 2003.

[162] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[163] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *18th Annual Conf. on Neural Information Processing Systems*, 2003.

[164] Y. Zhou, T. Zhang, and J. Sun. Music style classification with a novel bayesian model. In *Advanced Data Mining and Applications*. Springer, 2006.

[165] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

VITA

BO SHAO

1970	Born, Yancheng, Jiangsu, P. R. China
1992	B.S., Mining Engineering Northeastern University Shenyang, P. R. China
1995	M.S., Computer Sciences and Applications Southeast University Nanjing, P. R. China
2008-2011	Doctoral Candidate in Computer Science Florida International University Miami, FL, USA

PUBLICATIONS
Book Chapters

- Tao Li, Mitsunori Ogihara, Bo Shao, and Dingding Wang. Machine Learning Approaches for Music Information Retrieval. In M. J. Er and Y. Zhou, eds., Theory and Novel Applications of Machine Learning, In-Tech Education and Publishing, 2009, ISBN 978-953-7619-55-4.

Journal Papers

- Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara. Music Recommendation Based on Acoustic Features and User Access Patterns. IEEE Transactions on Audio, Speech and Language Processing, 17(8):1602-1611, 2009.

- Tao Li, Mitsunori Ogihara, Wei Peng, Bo Shao, and Shenghuo Zhu. Music Clustering with Features from Different Information Sources. IEEE Transactions on Multimedia, 11(3): 477-485, 2009.

Conference/Workshop Publications

- Bo Shao, Tao Li, and Mitsunori Ogihara. Quantify Music Artist Similarity Based on Style and Mood. In Proceedings of 10th ACM workshop on Web information and data management (WIDM 2008), Pages 119-124, 2008.

- Fei Wang, Xin Wang, Bo Shao, Tao Li, and Mitsunori Ogihara. Tag Integrated Multi-Label Music Style Classification with Hypergraph. In Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009), Pages 363-368, 2009.

- Wei Peng, Tao Li and Bo Shao. Clustering Multiway Data via Adaptive Subspace Iteration. In Proceedings of the Conference on Information and Knowledge Management (CIKM 2008), Pages 1519-1520, 2008.

- Tao Li, Chris Ding, Yi Zhang, and Bo Shao. Knowledge Transformation from Word Space to Document Space. In Proceedings of The 31st Annual International ACM SIGIR Conference (SIGIR 2008), Pages 187-194, 2008.