UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DEI
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE

SCUOLA DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE
INDIRIZZO IN SCIENZA E TECNOLOGIA DELL'INFORMAZIONE, CICLO: XXIII

Content-based Music Access: Combining Audio Features and Semantic Information for Music Search Engines

Direttore della Scuola
CHIAR.MO PROF. MATTEO BERTOCCO

Supervisore
DR. NICOLA ORIO

Dottorando
ING. RICCARDO MIOTTO

31 GENNAIO 2011

To my family and friends.

Abstract

During the last decade, the Internet has reinvented the music industry. Physical media have evolved towards online products and services. As a consequence of this transition, online music corpora have reached a massive scale and are constantly being enriched with new documents. At the same time, a great quantity of cultural heritage content remains undisclosed because of the lack of metadata to describe and contextualize it. This has created a need for music retrieval and discovery technologies that allow users to interact with all these music repositories efficiently and effectively.

Music Information Retrieval (MIR) is the research field that studies methods and tools for improving such interaction as well as access to music documents. Most of the research works in MIR focuses on content-based approaches, which exploit the analysis of the audio signal of a song to extract significant descriptors of the music content. These content descriptors may be processed and used in different application scenarios, such as retrieval, recommendation, dissemination, musicology analysis, and so on. The thesis explores novel automatic (content-based) methodologies for music retrieval which are based on semantic textual descriptors, acoustic similarity, and a combination of the two; we show empirically how the proposed approaches lead to efficient and competitive solutions with respect to other alternative state-of-the-art strategies.

Part of the thesis focuses on music discovery systems, that is search engines where users do not look for a specific song or artist, but may have some general criteria they wish to satisfy. These criteria are commonly expressed in the form of tags, that is short phrases that capture relevant characteristics of the songs, such as genre, instrumentation, emotions, and so on. Because of the scale of current collections, manually assigning tags to songs is becoming an infeasible task; for this reason the automatic tagging of music content is now considered a core challenge in the design of fully functional music retrieval systems. State-of-the-art content-based systems for music annotation (which are usually called auto-taggers) model the acoustic patterns of the songs associated with each tag in a vocabulary through machine learning approaches. Based on these tag models, auto-taggers generate a vector

of tag weights when annotating a new song. This vector may be interpreted as a semantic multinomial (SMN), that is a distribution characterizing the relevance of each tag to a song, which can be used for music annotation and retrieval.

A first original contribution reported in the thesis aims at improving state-of-the-art auto-taggers by considering tag co-occurrences. While a listener may derive semantic associations for audio clips from direct auditory cues (e.g. hearing "bass guitar") as well as from context (e.g. inferring "bass guitar" in the context of a "rock" song), auto-taggers ignore this context. Indeed, although contextual relationships correlate tags, many state-of-the-art auto-taggers model tags independently. We present a novel approach for improving automatic music annotation by modeling contextual relationships between tags. A Dirichlet mixture model (DMM) is proposed as a second, additional stage in the modeling process to supplement any auto-tagging system that generates a semantic multinomial over a vocabulary of tags. For each tag in the vocabulary, a DMM captures the broader context defined by the tag by modeling tag co-occurrence patterns in the SMNs of songs associated with the tag. When annotating songs, the DMMs refine SMN annotations by leveraging contextual evidence. Experimental results demonstrate the benefits of combining a variety of auto-taggers with this generative context model; it generally outperforms other approaches to context modeling as well.

The use of tags alone allows for efficient and effective music retrieval mechanisms; however, automatic tagging strategies may lead to noisy representations that may negatively affect the effectiveness of retrieval algorithms. Yet, search and discovery operations across music collections can be also carried out matching users interests or exploiting acoustic similarity. One major issue in music information retrieval is how to combine such noisy and heterogeneous information sources in order to improve retrieval effectiveness. At this aim, the thesis explores a statistical retrieval framework based on combining tags and acoustic similarity through a hidden Markov model. The retrieval mechanism relies on an application of the Viterbi algorithm which highlights the sequence of songs that best represents a user query. The model is presented for improving state-of-the-art music search and discovery engines by delivering more relevant ranking lists. In fact, through an empirical evaluation we show how the proposed model leads to better performances than retrieval approaches which rank songs according to individual information sources alone or which use a combination of them. Additionally, the high generality of the framework makes it suitable for other media as well, such as images and videos.

Besides music discovery, the thesis challenges also the problem of music identification, the goal which is to match different recordings of the same songs (i.e. finding covers of a given query). At this aim we present two novel music descriptors based on the harmonic content of the audio signals. Their main purpose is to provide a compact representation which is likely to be shared by different performances of the same music score. At the same time, they also aim at reducing the storage requirements of the music representation as well as enabling efficient retrieval over large music corpora. The effectiveness of these two descriptors, combined in a

single scalable system, has been tested for classical music identification, which is probably the applicative scenario that mostly needs automatic strategies for labeling unknown recordings. Scalability is guaranteed by an index-based pre-retrieval step which handles music features as textual words; in addition, precision in the identification is brought by alignment carried out through an application of hidden Markov models. Results with a collection of more than ten thousand recordings have been satisfying in terms of efficiency and effectiveness.

Sommario

Nell'ultimo decennio l'avvento di Internet ha reinventato l'industria musicale, in particolare i supporti fisici si sono evoluti verso prodotti e servizi reperibili online. Questa transizione ha portato le collezioni musicali disponibili su Internet ad avere dimensioni enormi e in continua crescita, a causa del quotidiano inserimento di nuovo contenuto musicale. Allo stesso tempo, una buona parte dei documenti musicali tipici del patrimonio culturale rimane inaccessibile, a causa della mancanza di dati che li descrivano e li contestualizzino. Tutto ciò evidenzia la necessità di nuove tecnologie che permettano agli utenti di interagire con tutte queste collezioni musicali in modo effettivo ed efficiente.

Il reperimento d'informazioni musicali (i.e. MIR) è il settore di ricerca che studia le tecniche e gli strumenti per migliorare sia questa interazione, sia l'accesso ai documenti musicali. La maggior parte della ricerca effettuata nel MIR riguarda tecniche automatiche basate sul contenuto (i.e. content-based), le quali analizzano il segnale audio di una canzone ed estraggono dei descrittori, che ne caratterizzano, appunto, il contenuto. Questi descrittori possono essere elaborati ed utilizzati in varie applicazioni: motori di ricerca, divulgazione, analisi musicologa e così via. La tesi presenta dei modelli originali content-based per motori di ricerca musicali di vario genere, che si basano, sia su descrittori semantici testuali e su similarità acustica, sia su una loro combinazione. Attraverso esperimenti pratici, dimostreremo come i modelli proposti ottengano prestazioni efficienti e competitive se confrontate con alcuni dei sistemi alternativi presenti nello stato dell'arte.

Una buona parte della tesi si concentra sui sistemi di *music discovery*, ovvero motori di ricerca nei quali gli utenti non cercano una canzone o un'artista specifico, ma hanno perlopiù un criterio generale che vogliono soddisfare. Questi criteri di ricerca sono in genere espressi sottoforma di *tag*, ovvero annotazioni che caratterizzano gli aspetti rilevanti delle canzoni (e.g. genere, strumenti, emozioni). A causa delle dimensioni raggiunte ormai dalle varie collezioni, l'assegnazione manuale dei tag alle canzoni è però diventata un'operazione impraticabile. Per questa ragione, i modelli che assegnano i tag in modo automatico sono diventati dei punti chiave nella progettazione dei motori di ricerca musicale. I sistemi content-based per

l'assegnazione automatica di tag (i.e. auto-tagger) generalmente si basano su approcci di machine learning, che modellano le caratteristiche audio delle canzoni associate ad un certo tag. Questi modelli sono poi utilizzati per annotare le nuove canzoni generando un vettore di pesi, uno per ogni tag nel vocabolario, che misurano la rilevanza che ogni tag ha per quella canzone (i.e. SMN).

Un primo contributo originale della tesi ha l'obiettivo di migliorare lo stato dell'arte degli auto-tagger, modellando le co-occorrenze tra i tag. Infatti mentre una persona può associare tag a una canzone sia direttamente (e.g. ascolta lo strumento"basso"), sia dal contesto (e.g. intuisce"basso" sapendo che la canzone è di genere "rock"), gli auto-tagger diversamente ignorano questo contesto. Infatti, nonostante le relazioni contestuali correlino i tag, la maggior parte degli auto-tagger modella ogni tag in modo indipendente. Il nostro sistema pertanto cerca di migliorare l'assegnazione automatica di tag, modellando le relazioni contestuali che occorrono tra i vari tag di un vocabolario. Per far questo utilizziamo un modello di misture di Dirichlet (DMM) al fine di migliorare qualsiasi auto-tagger che genera delle SMN. Per ogni tag nel vocabolario, una DMM è usata per catturare le co-occorrenze con gli altri tag nelle SMN delle canzoni associate con quel tag. Quando una nuova canzone è annotata, il DMM rifinisce le SMN prodotte da un auto-tagger sfruttando le sue caratteristiche contestuali. I risultati sperimentali dimostrano i benefici di combinare vari auto-tagger con le DMM; in aggiunta, i risultati migliorano rispetto anche a quelli ottenuti con modelli contestuali alternativi dello stato dell'arte.

L'uso dei tag permette di costruire efficienti ed effettivi motori di ricerca musicali; tuttavia le strategie automatiche per l'assegnazione di tag a volte ottengono rappresentazioni non precise che possono influenzare negativamente le funzioni di reperimento. Al tempo stesso, le ricerca di documenti musicali può essere anche fatta confrontando gli interessi degli utenti o sfruttando le similarità acustiche tra le canzoni. Uno dei principali problemi aperti nel MIR è come combinare tutte queste diverse informazioni per migliorare le funzioni di ricerca. Ponendosi questo obiettivo, la tesi propone un modello di reperimento statistico basato sulla combinazione tra i tag e la similarità acustica mediante un modello di Markov nascosto. Il meccanismo di ricerca si basa su un'applicazione dell'algoritmo di Viterbi, il quale estrae dal modello la sequenza di canzoni che meglio rappresenta la query. L'obiettivo è di migliorare lo stato dell'arte dei sistemi di ricerca musicale e, in particolare, di music discovery fornendo all'utente liste di canzoni maggiormente rilevanti. Gli esperimenti infatti mostrano come il modello proposto risulta migliore sia di algoritmi che ordinano le canzoni utilizzando un'informazione sola, sia di quelli che le combinano in modo diverso. In aggiunta, l'alta generalità del modello lo rende adatto anche ad altri settori multimediali, come le immagini e i video.

In parallelo con i sistemi di music discovery, la tesi affronta anche il problema di identificazione musicale (i.e. music identification), il cui obiettivo è quello di associare tra loro diverse registrazioni audio che condividono lo stesso spartito musicale (i.e. trovare le versioni cover di una certa query). In funzione di questo, la tesi presenta due descrittori che si basano

sulla progressione armonica della musica. Il loro scopo principale è quello di fornire una rappresentazione compatta del segnale audio che possa essere condivisa dalle canzoni aventi lo stesso spartito musicale. Al tempo stesso, mirano anche a ridurre lo spazio di memoria occupato e a permettere operazioni di ricerca efficienti anche in presenza di grandi collezioni. La validità dei due descrittori è stata verificata per l'identificazione di musica classica, ovvero lo scenario che maggiormente necessita di strategie automatiche per la gestione di registrazioni audio non catalogate. La scalabilità del sistema è garantita da una pre-ricerca basata su un indice che gestisce i descrittori musicali come fossero parole di un testo; in aggiunta, la precisione dell'identificazione è aumentata mediante un'operazione di allineamento eseguita utilizzando i modelli di Markov nascosti. I risultati sperimentali ottenuti con una collezione di più di diecimila registrazioni audio sono stati soddisfacenti sia da un punto di vista di efficienza sia di efficacia.

Acknowledgements

This thesis would not have been possible without the immense support that I received during the course of my PhD.

Firstly, I would like to acknowledge my supervisor Dr. Nicola Orio for giving me the opportunity of being part of the Information Management System group back in 2007, and the guidance and confidence to pursue this work.

In addition, I thank Professors Maristella Agosti and Massimo Melucci, and Dr. Nicola Ferro for their suggestions and discussions about the wide area of information retrieval as well as for all the collaborations carried out in the past years. The people with whom I have shared an office with over the last three years, such as Gianmaria Silvello, Emanuele Di Buccio, Marco Dussin, Ivano Masiero, Nicola Montecchio, and Giorgio Maria Di Nunzio, have provided me with a stimulating research and enjoyable environment. To all of them also goes my gratitude.

Research on automatic music tagging was supervised by Professor Gert Lanckriet during the period I spent as visiting scholar at University of California, San Diego, USA. His suggestions and attention to details have enabled the work in Chapter 4 to flourish. So I acknowledge him along with the rest of the Cal Audition Laboratory crew: Luke Barrington, Brian McFee, and Emanuele Coviello. The discussions with all of them on music search engines and tags had a strong impact on this thesis. A special thanks goes to Nikhil Rasiwasia as well for the helpful discussions and suggestions on modeling tag co-occurrences.

Last but not least, I would like to thank my parents, Renzo and Carla, and my brother, Tommaso, whose love and support made it possible for me to complete this work. Additionally, I would like to acknowledge my dog, Lucky, faithful companion of all these years.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The rise of the Internet and the new technologies have changed the way music is produced and consumed. A musician in South America with a basic desktop computer, an inexpensive microphone and free audio editing software can record and produce reasonably high-quality music. He can post his songs on any number of musically-oriented social networks making them accessible to the public. A music consumer in Italy can then rapidly download these songs over high-bandwidth Internet connection and store them on a personal MP3 player.

The music industry is facing these changes as well: CD sales are declining, while the online stores and the new markets around the legal downloading of music are continuously increasing both in terms of sales and songs. As an example, CD sales in United States have dropped from 147 million in 2009 to 114 million in 2010; conversely, about 44 million digital albums have been sold in 2010, 4 millions more than the same period in 2009[1].

As a result, *millions of songs are now instantly available to millions of people*. While this current age of music proliferation provides new opportunities for producers (e.g. artists) and consumers (e.g. listeners), it also creates a need for novel music search and discovery technologies.

Available commercial systems generally access and manage music solely on the basis of manual annotations, in particular metadata (i.e. song title, artist, album title, year, record label, etc.) and *tags*. Tags are short phrases of one to about three word aimed at describing digital items; in the case of music they may define the genre, the instrumentation, the evoked mood, the usage a song is particularly suitable for, and so on. In any case, the manual annotation of music content is a time consuming and error-prone task. In addition, as also happens with other media, different individuals tend to tag songs using different conventions,

[1] In this chapter, we report numerical details about music technologies, in particular concerning history, sales, users, and songs. Generally, these have been gathered from the corresponding music industry websites and Billboard articles (`http://www.billboard.com/`).

languages, and forms. These limitations are more accentuated by the continuous release of new music and the size of present-day music corpora; most of the new content could be without annotations for a long period.

At the same time, while the amount of new music content explodes, a great deal of cultural heritage content of great interest remains undisclosed. For instance, theatres, concert halls, radio and television broadcasters have usually hundreds of hours of almost unlabeled analog recordings, which witness the activities over the years of the institution and that need to be digitized and catalogued for preservation and dissemination. Manual identification aimed at cataloguing music works is a difficult task that, ideally, should be carried out by trained users who remember by heart hundreds, or even thousands, of hours of music. This task is even more difficult with instrumental music, where lyrics are not available for recognizing a particular work.

These and other problems call for automatic solutions to analyze, describe and index music items. This thesis focuses on automatic approaches based on descriptors of the music content for building unsupervised (or semi-supervised) music retrieval systems which does not rely (or little rely) on human efforts. The music descriptors are obtained by processing the audio signal of the songs in order to extract relevant features describing the characteristics of the music. These music descriptors can be used in several applications, such as to compute acoustic similarity between songs, to automatically tag songs, to identify an unknown piece of music, and so on. Along the dissertation we will generally refer to the these automatic approaches as *content-based*.

In this chapter, we discuss how technology is changing the music access, and briefly overview some existing commercial systems. All the highlighted limitations point out the need for content-based approaches to improve and integrate state-of-the-art music retrieval systems. We then summarize the content and the structure of the thesis, particularly highlighting its original contributions.

1.1 Music Information Access

The explosion of music content can be explained by different reasons. First of all, the characteristics of music language itself. Music is an art form that can be shared by people with different culture because it crosses the barriers of national languages and cultural backgrounds. For example, Western music has passionate followers in Japan, and many persons in Europe are keen on Indian music: all of them can enjoy music without the need of a translation, which is normally required for accessing foreign textual works. Another reason is that technology for music recording, digitization, and playback allows users for an access that is almost comparable to the listening of a live performance, at least at the level of audio quality; in addition the signal-to-noise ratio is better for digital formats than for many analog formats. This is not the case with other art forms, like paintings or sculptures, for which the digital version is only an approximate representation of the artwork. Indeed, the access to

digitized paintings can be useful for studying the works of a given artist but cannot substitute the direct interaction with the original version. Additionally, the increasing interest toward digital music may be motivated by its portability as well as by the possibility to access music while doing another, possibly primary, activity (e.g. sport, travelling, sleeping).

Technology plays a key role in this music explosion as well, mainly because it facilitates the production and the distribution of music. In this section, we focus on the role played by technology in this music proliferation age; in particular, we discuss aspects related to the production of music, the online stores where it is possible to get physically the music (i.e. downloads), and the last frontier of only-consumption systems (i.e. Web radios and streaming services).

1.1.1 Production

In the early 1990s, the compact disc (CD) replaced the cassette tape as the leading medium for music distribution. CDs are characterized by a smaller size, high-fidelity digital format, lack of deterioration with playback, and improved functionality (e.g. skipping/replaying songs). Additionally, they can be played with almost each personal computer, while tapes can not. At that time, there was a significant cost associated with producing an album: recording the music in a studio, mixing the raw audio on an expensive sound board, producing a digital master copy, and so on. In order, to make this process financially profitable, an artist (or record label) would have to sell hundreds of thousands of CDs.

Today, the production process has changed in many ways and to produce a CD is now easier and more economic. In fact, almost every personal computer comes equipped with a sound card, a microphone, and a CD-burner. Additionally, mixing boards can be emulated using different software packages that allow for recording and editing sounds, such as GarageBand which comes with every Apple computer, and Audacity which is free, open source and available for all the most common operating systems.

Nevertheless, CD sales are rapidly decreasing. This mostly depends by the fact that compressed audio files (e.g. MP3, AAC), high-bandwidth Internet connections, inexpensive hard disks, and online music repositories have eliminated the need for the physical storage of music on CDs. In any case, as final result, since these low production costs, an amateur musician has now fewer barriers to join a market that was once an exclusive domain of the big record companies.

1.1.2 Online Stores

Just as computers have affected the music production, the Internet has changed the way music is distributed. Online legal (and illegal) virtual stores represent nowadays the main way to obtain songs and albums[2]. An online music store is an online business which provides

[2]On April 3, 2008 iTunes Store surpassed Walmart as the biggest music retailer in the US, a milestone in the music industry as it is the first time in history that an online music retailer exceeds those of physical

audio files, usually music, on a per-song and/or subscription basis. The music store offers the actual music file, that can be bought and downloaded by the users in their own computer.

In the late 1990s, peer-to-peer (P2P) file sharing services became the first popular way to (illegally) distribute music content over the Internet. The most famous P2P company was Napster, which operated between June 1999 and July 2001. Its technology allowed people to easily share their MP3 files with other participants, bypassing the established market for such songs and thus leading to massive copyright violations of music and film media as well as other intellectual property. After peaking with 1.5 million simultaneous users, Napster was shut down in the summer of 2001 by a lawsuit over copyright infringement. However, a number of P2P networks have followed Napster and continue to flourish despite the constant threat of legal action from the recording industry.

One of the first digital legal store has been iTunes, launched by Apple in 2003. In order to avoid illegal uses of the sold content, Apple began selling songs that were protected by a digital rights management (DRM) copyright protection system. DRM limited iTunes music to the Apples iPod portable music player, placed limits on how many times the music could be copied onto different computers, and made it difficult to recode the music into other (non-DRM) formats. However, competitors accused Apple of using the DRM to establish a vertical monopoly and a lock-in for iPod users to use the iTunes Store exclusively (and vice versa). This led to several disputes with the antitrust agencies; therefore Apple initiated a shift into selling free-DRM music in most countries, achieving the complete removal of DRM on all the songs of the collection for the US market in April 2009. Nowadays, iTunes accounts for 70% of worldwide online digital music sales in 2010, making the service the largest legal music retailer. At the end of February 2010, the store served its 10 billionth song download; this major milestone was reached in just under seven years of being online; currently, the iTunes store contains about 14 million songs.

As a result of Apples success, numerous companies have entered the market with competitive prices and non-DRM music. In particular, eMusic[3] was one of the first sites to sell music in a free-DRM MP3 format; prior to July 2009, eMusic sold only music from independent artists[4], which is generally characterized by non-mainstream music genres (e.g. underground, experimental music, new age). Since early 2010, eMusic has started to strike deals with some of the big companies (i.e. Sony Music Entertainment and Warner Music) to sell also their music; as result, the store is rapidly increasing and now contains about 6 million songs. Lastly, other alternative stores are Jamendo and Amazon.

music formats.

[3] `http://www.emusic.com/`

[4] An *independent artist* is an artist that is not signed by one of the four major music companies: Sony BMG Music Entertainment, Warner Music Group Corporation, Universal Music, and EMI.

1.1.3 Online Consumption

Besides the online music stores, in the last years streaming services such as personalized radios have represented the new frontier of music consumption, together with social networks. Streaming services offer full (or partial) listening of the music content to the users via Internet, without releasing the audio files (i.e. the users do not own the source file but can just listen to it). YouTube is probably the most famous streaming service to freely (but potentially illegally) access music videos that are mostly posted by users.

All the major radio broadcasters provide Web radio services (e.g. Virgin Radio), which consist of continuous streams of music representing a certain criteria, such as a genre (e.g. rock, disco), a period (e.g. 80s, Christmas period), a country (e.g. Italian, Indian), and so on; iTunes, the Apple music player interfaced with iTunes Store, offers this service as well. Users appreciate Web radios because they can listen to and discover new music legally, without incurring in copyright problems.

Differently, on-demand music access services allow users to generate their own radio. Pandora[5] revolutionized the Internet radio by offering personalized streams of recommended music. Users enter a song or artist that they enjoy, and the service responds by playing selections that are musically similar. Music similarity is based on the analysis of human experts who annotate songs using a set of 400 music concepts (referred as *Music Genome* in market literature). The simplicity of the user interface, as well as the quality of the music recommendations have resulted in 48 million users registered in March 2010; however, the size of the collection is limited to about 700,000 tracks only. In alternative, Spotify[6] is a more recent on-demand music systems (launched in 2008), offering music of both major and independent record labels. Users can register either for free accounts supported by visual and radio-style advertising or for paid subscriptions without ads and with a range of extra features such as higher bitrate streams and offline access to music. In September of 2010, the company has declared ten million users, with approximately 650,000 of them that are paying members; the collection is declared to contain more than 10 million songs.Other major Internet companies include Yahoo Launchcast, Slacker, and AccRadio.

Social networks represent the last Internet development that is changing how music is consumed. Myspace, which was launched in 2003, is a network of music artists, where they can promote themselves and upload their music (approximatively 5 million artist pages in 2008). It generally includes music streaming, music merchandise and concert tickets. In 2009 MySpace also acquired iLike, an alternative online service that allows users to download and share music which became famous for its integration with Facebook. As of June 2010, MySpace has about 66 million users. In alternative, Last.fm[7] is a music-specific social network launched in 2002 that depends on its users to generate Web content: artist biographies and album reviews are created using a public wiki-like form, popularity charts are based on user

[5] `http://www.pandora.com/`
[6] `http://www.spotify.com/`
[7] `http://www.last.fm/`

listening habits, song and artist recommendations are based on collaborative filtering, and social tags are collected using a text box in the audio player interface. As of March 2009, Last.fm reported that its database contained about 7 million songs from artists on all the major commercial and independent labels, with approximately 40 million users. Users are not allowed to upload copyrighted audio files but commercially available albums are regularly added by Last.fm staff; conversely, labels and artists can upload their own music content for streaming. To the best of our knowledge, the last social network about music released is Ping (September 2010); its major feature is the integration with iTunes.

Besides the Web-base interface, all these services are generally available also for the phone markets, especially iPhone and Android-based devices. Additionally, it should be noted that many music services are not freely available in Italy. For example, Last.fm is not free as in UK and in US; in fact, after a trial of 30 tracks listened, a quote of 3€ per month has to be paid to keep using the service. Similarly, Spotify is available in the paying version only. Lastly, Pandora does not work at all in our country (and within the European Union in general), mainly because of copyright issues.

1.2 Thesis and Context

All the systems and services reviewed in the previous section use metadata and tags for music retrieval. In addition, they sometimes exploit collaborative filtering techniques that analyze a huge amount of user behaviors for delivering recommendations that consider also user similarity. As an example, collaborative filtering is extensively used by Genius, a service integrated with iTunes, which recommends songs from the user personal library and the iTunes store with respect to the similarity to a chosen seed song; Last.fm uses collaborative filtering as well. The large use of metadata and tags is easily explained considering that the use of text facilitates the interaction between human-users and music technologies; in fact, common users of music engines are more comfortable in expressing their requirements with descriptive words rather than using musical terms (e.g. note and chord names, time signatures, musical forms).

However, the large scale of nowadays corpora and the continuous release of new music rise several limitations. First, high-quality human-based annotations do not scale well with this massive release of music. For instance, the Pandora dataset is much smaller that the other systems (700,000 songs against the 6 and 10 million songs of Last.fm and Spotify respectively); this is also due by the fact that the analysis performed on each song by its musicologists is a very time-consuming task (i.e. about 20 - 30 minutes per song). Second, social tags typical of Last.fm may be error-prone, noisy, and inconsistent because assigned by non-expert users, wildly and without restrictions; additionally they may also be in different languages. Third, collaborative filtering applies only to existing user behaviors, meaning that if a song is without tags, or has never been recommended, it can not be included in the analysis performed by the retrieval algorithms. This is generally referred as the *cold*

start problem of recommendation systems [1]. Lastly, collaborative filtering tends to favour famous songs and artists that are largely played by the greatest part of the users, despite the huge amount of content released by less known artists. This leads to systems that generally recommend only a little subset of all the songs in the collection. This effect is known as the *long tail* of recommendation systems [21]. The long tail distribution is composed by a small number of popular items (i.e. the hits), while all the rest located in the tail of the curve (i.e. the less known songs). While the hits are commonly recommended by music systems, the documents in the tail tend to remain hidden. The goal of the long tail market is to include in the recommendations the documents in the tail as well (i.e. discovery of the long tail); an example of long tail distribution is depicted in Figure 1.1[8].

The use of content-based approaches to automatically process the information carried by the audio signal can be used to deliver more efficient solutions and to partially solve these limitations. In fact, the audio content may be used to automatically tag songs, to deliver objective similarity, to identify unknown recordings, to detect copyright infringements, as well as for more specific tasks such as alignment, chord recognition, and so on. While content-based approaches are very objective and cannot well express the high subjectivity of the music similarity perception (i.e. they cannot substitute completely the humans), they can provide tools that integrate text-based commercial systems in order to provide the users with better services. For instance, the use of a content-based automatic tagging systems allow for an immediate annotations of the new released songs, which would help reducing the cold start problem, as well as improving the discovery of the long tail. Alternatively, content-based music identification systems can help the user in charge of digitalizing and cataloguing all the analog recordings stored by a theatre or by a radio broadcaster.

1.2.1 Original Contributions

The thesis presents some novel methodologies for content-based music retrieval; in particular, the original contributions reported are:

- a novel approach for the automatic tagging of music content. Considering that state-of-the-art systems assign tags according to acoustic patterns in the audio content only, we propose to model also tags correlation in order to leverage the tags prediction. The purpose is to provide a reliable technique to automatically tag music, which could be used to efficiently and effectively annotate the new music content released; the details of the approach are reported in Chapter 4;

- a novel approach for music retrieval that use a statistical model to combine tags and acoustic similarity in a single retrieval framework. The system responds to user-queries composed either by a combination of tags or by a song and return a list of songs ranked by similarity with the query. The purpose is to provide a retrieval model that combines

[8] We extracted the picture in Figure 1.1 from: O. Celma, "Music Recommendation and Discovery in the Long Tail" [21].

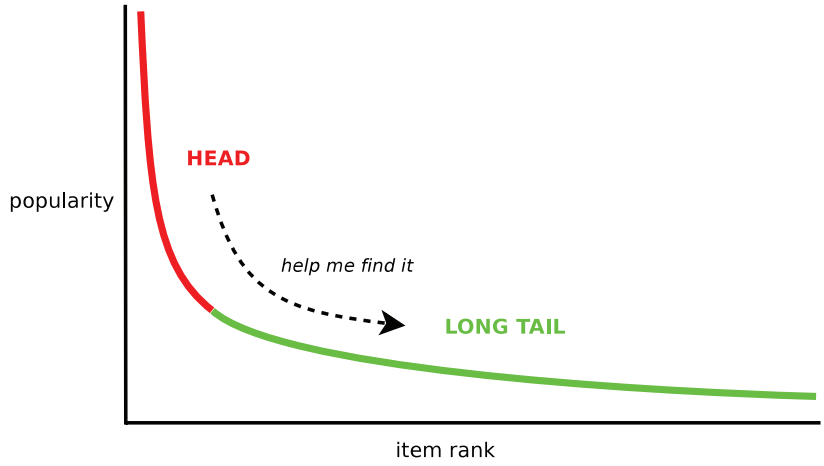


Figure 1.1. The Long Tail of items in a recommender systems: while highly popular items are also highly recommended, there is a huge quantity of less-known documents that are little ranked by state-of-the-art algorithms.

both objective similarity (i.e. acoustic) and subjective descriptions (i.e. tags) for a better discovery of the long tail; details of the model are in Chapter 5;

- two novel compact music descriptors which have been optimized for automatic identification of unknown audio recordings. Details about the features are reported in Appendixes A and B; additionally an overview of the identification system we built for classical music using these novel descriptors is reported in Section 2.

1.2.2 Structure of the Thesis

The thesis mostly focuses on the original research carried out for music search and discovery systems; early work on music identification is therefore treated as a complement of the main work. As a consequence, the thesis is organized as follow. Chapter 2 overviews the music information retrieval research area, by particularly focusing on content-based automatic approaches. Among them, we give particular attention to the music identification system previously mentioned, referring the reader to the appendixes for the details about the novel audio features proposed for the task. Chapters 3 presents the thesis background, that is algorithms and instruments used to develop and test our models. The original contributions for automatic tagging and music retrieval and the empirical evaluation performed are reported in Chapters 4, 5, and 6 respectively. Lastly, conclusions about content-based music systems and where they could drive the music industry are discussed in Chapter 7.

CHAPTER 2

Content-based Music Information Retrieval

Over the last decade the research field of music information retrieval (MIR) has encountered a remarkable gain of attention because of the continuous increasing of digital music available. MIR devotes to fulfill users music information needs, encompassing a number of different approaches aimed at music management, easy access, and enjoyment. Most of the research work on MIR, of the proposed techniques and of the developed systems are *content-based*. The main idea underlying content-based approaches is that a document can be described by a set of features that are directly computed from its content; the basic assumption is that metadata and tags are either not suitable, or unreliable, or missing. In the case of music, as already introduced in Chapter 1, all the three assumptions may hold. In fact, music textual descriptors may not be suitable because either too generic or specific, and unreliable because there is no control on how the information has been added; additionally they can be completely missing. It is interesting to note that both CDs and first MP3 audio standards, which gave rise to the two digital revolutions of music enjoyment, do not take into account the possibility of carrying also structured metadata information. The possibility to include unstructured textual information in MP3 was introduced in a later version; however, fields are not mandatory, and it is up to the person who creates the MP3 to spend time adding this information.

This chapter overviews some content-based MIR tasks, with the purpose of briefly introducing an unfamiliar reader to the research area. In addition, we present an original work that we carried out for the task of music identification. Yet, we start introducing common concepts and terminology frequently used along the thesis.

2.1 Music Concepts and Terminology

Most of the approaches and techniques for music retrieval are based on a number of music concepts that may not be familiar to persons without musical training. For this reason, this section presents a short introduction of some basic concepts and terms[1].

Each musical instrument, with the exception of some percussions, produces almost periodic vibrations. In particular, the sounds produced by musical instruments are the result of the combination of different frequencies, which are all multiple integers of a *fundamental frequency*. The three basic features of a musical sound are [73]:

- **pitch**, which is related to the perception of the fundamental frequency of a sound; pitch is said to range from *low* or *deep* to *high* or *acute* sounds;
- **intensity**, which is related to the amplitude, and thus to the energy, of the vibration; textual labels for intensity range from *soft* to *loud*; the intensity is also defined *loudness*;
- **timbre**, which is defined as the sound characteristics that allow listeners to perceive as different two sounds with same pitch and intensity.

The perception of pitch and intensity is not as straightforward as the above definitions may suggest, since the human ear does not have a linear behavior neither in pitch recognition nor in the perception of intensity. Yet, these two perceptually relevant qualities of sounds can be reasonably approximated considering the fundamental frequency and the energy of a sound. Timbre, on the other hand, is a multidimensional sound quality that is not easily described with simple features.

When two or more sounds are played together, they form a *chord*. A chord may have different qualities, depending on the pitch of the different sounds and, in particular, on the distances between them. Chords play a fundamental role in many music genres (e.g. pop, rock, jazz), where polyphonic musical instruments (e.g. piano, keyboard, guitar) are often dedicated to the accompaniment and basically play chords. A sequence of chords defines the *harmonic* content of a song.

Besides these main basic concepts, other terms derived from music theory and practice are generally used to describe music within the MIR community. In particular:

- the **tempo** is the speed at which a musical work is played, or expected to be played, by performers. The tempo is usually measured in beats-per-minute;
- the **tonality** of a song is related to the role played by the different chords of a musical work; tonality is defined by the name of the chord that plays a central role in a musical work. The concept of tonality may not be applicable to some music genres;

[1] Being very simple, these operative definitions may not be completely agreed by readers with a musical background, because some relevant aspects are not taken into account. Yet, their purpose is just to introduce some terminology that will be used in the next chapters.

- the **time signature** gives information on the organization of strong and soft beats along the time axis;
- the **key signature** is an incomplete representation of the tonality, which is useful for performers because it expresses which are the notes that have to be consistently played altered.

Other concepts are more related to the production of sounds and to the parameters that describe single or groups of notes. In particular, a note starts being perceived at its *onset time*, lasts for a given duration, and stops to be perceived at its *offset time*. Additionally, a sequence of note generally indicates the *melody* of a song. Lastly, the sounds are produced by musical instruments and by the voice; depending on their confirmation, these can produce a limited range of pitches, which is generally called instrument (voice) *register*.

2.2 Overview of Content-based MIR Research Tasks

From a general point of view MIR is concerned with the extraction, analysis, and usage of information about any kind of music entity (e.g. a song or an artist). As regards a song, we mostly discriminate between the score, which is the symbolic description of a piece of music (i.e. generally stored as MIDI format) and the audio recording. The latter is the audio signal that results by a sequence of gestures performed by musicians on their musical instruments when playing a score. Audio recordings are widely digitally stored as WAV, MP3 and AAC format. The following sections are an attempt to group some of the heterogeneous research areas within the field of MIR. However, we dot claim to exhaustively cover all the areas, but rather to point out some of the important subfields of MIR concerning information retrieval and audio recordings processing[2]. In addition, we give more attention to the task of classical music identification.

2.2.1 Feature Extraction

Extraction of meaningful features from the audio signal is probably the most important MIR task. Extracting good features is far from easy and involves techniques and theories from signal processing, psychoacoustic, music perception, and data analysis. Audio signal-based descriptors are frequently categorized into low and high level features. In general, low-level descriptors are closely related to the audio signal: examples are energy in different frequency bands, zero crossing rate of the waveform, spectral centroid, or spectral flux. Differently, high-level features represent more abstract properties of music such as intensity, tempo, melody, pitch, harmony, tonality, and timbre. A complete discussion on content-based audio features has been reported by Pampalk [74]. Several open-source packages have been publicly released to extract features form the music content, in particular Marsyas [103] and the

[2]There is a wide branch in MIR which analyses and processes the symbolic representation, mostly for musicology analysis. However, this research is out of the scope of the dissertation and we do not consider it.

MIRtoolbox [56]. Additionally, closed-source online services that provide the users with content descriptors are now available, such as The Echo Nest[3].

Appendixes A and B present two original audio descriptors optimized for a music identification task (see Section 2.2.6). Both features process the pitch of the audio signal in order to achieve a general representation that could be shared by different audio recordings of the same music score. In addition, Section 3.1 reports two algorithms for extracting the timbral descriptors from an audio signal.

2.2.2 Acoustic Similarity

Acoustic similarity is the task of using audio features to compute similarity relationships between songs. Acoustic similarity is *objective* in the sense that it does not consider the subjectivity of music perception. Additionally, not any similarity measure may give worthwhile results. Hence, the choice of a good combination of features and similarity measures is crucial when designing MIR applications.

The application context defines which features are of interest. For example, to generate better than random playlist does not necessarily require knowledge of the melody or chords but would definitely benefit from knowledge concerning the tempo or instrumentation. Conversely, the identification of different versions of a same music work requires to mostly focus on the pitch of the audio signal. Nevertheless, a number of the systems for computing acoustic similarity tend to include timbral descriptors, since timbre is a very common aspect shared by similar song.

In literature, early approaches for acoustic similarity used mixtures of Gaussians to represent the songs, and compute similarity as the divergence between the models. In particular, interesting results were achieved by Mandel and Ellis using a single Gaussian to model the timbre of the songs, and Kullback-Leibler (KL) divergence to compute similarity [60]. Recently alternative competitive algorithms have been proposed. Hoffman et al. [47] use the hierarchical Dirichlet process to automatically discover the latent structure within and across groups of songs, in order to generate a compact representation of each song, which could be efficiently compared using KL divergence. Alternatively, Seyerlehner et al. [86] represents songs through a non-parametric variant based on vector quantization; this representation allows for more powerful search strategies (i.e. KD-Trees, Locality Sensitive Hashing). In addition, the Music Information Retrieval Evaluation eXchange (MIREX)[4] has been running the music similarity task since 2005; many other content-based approaches can be found in the records of similar yearly campaigns.

[3] `http://the.echonest.com/`

[4] `http://www.music-ir.org/mirex/wiki/MIREX_HOME/`

2.2.3 Structural Analysis, Transcription, and Alignment

These research areas mostly concern the acoustic analysis of audio signals with the objective of revealing their structure, at different granularity levels. Structural analysis generally involves feature extraction and segmentation in first stages. Based on the results of the segmentation process, recurring acoustic patterns are sought, commonly by calculating self-similarities – i.e. similarities between the different segments of one and the same piece of music. Such self-similarities serve as indication for certain elements of a piece of music (e.g. verse, chorus, etc.). Applications of structural analysis include music summarization, generation of structural descriptors that facilitate the annotation of music pieces, and audio synthesis.

Related to structural analysis is the topic of *audio-to-score transcription*, which aims at recognizing individual notes or chords in an audio signal. While this can be considered a trivial task for monophonic music, it becomes harder when dealing with polyphonic material.

Lastly, the task of *audio-to-score alignment* refers to the ability of a system to automatically align an audio signal of a music performance with its score. More precisely, given a recording of a music performance and its score (defined as a sequence of musical events, such as notes, chords and rests, each characterized by a specified duration), the aim of a score alignment system is to match each sample of the audio stream with the musical event it belongs to; this is equivalent to finding the onset time, in the audio signal, of each score event. The audio-to-score alignment is also known as *score follower* when it is done in real time [71].

2.2.4 Music Tonality Analysis

This subfield of MIR is concerned with the extraction of tonal information from music. Tonality is related to the fundamental frequency of the audio signal as well as to its key signature. Within the MIR community, tonal descriptors are often named as chroma features, pitch class distributions or pitch class profiles. These terms are sometimes used to refer to the same concept, but they usually rely on very different implementations that try to obtain a similar result: a vector of features describing the different tones (or pitches) in an audio signal, excerpt, or frame. We used tonal descriptor to derive the audio feature described in Appendix A.

In Western music, the tonal system is composed of 12 semitones; for this reasons, tonal features are generally 12-dimensional vectors where each value expresses the amount of energy found in the audio for each semitone. Additionally, a finer analysis could be done, obtaining 24 or 36-dimensional tonal vectors (i.e. with an interval resolution of 1/2 and 1/3 of semitones respectively). In any case, according to [41], reliable pitch class distribution descriptors should generally fulfil the following requirements: (i) represent the pitch classes of both monophonic and polyphonic signals; (ii) consider the presence of harmonic frequencies; (iii) be robust to noise; (iv) be independent of timbre, played instrument, loudness, dynamics, and tuning. A

limitation of any tonal representation is the sensitivity to tonality shifts; this means that the same audio excerpt played in two different tonalities is characterized by two different tonal representations. Yet, the problem is generally tackled by considering that a change in the tonality simply corresponds to a rotation of some semitones in the tonal descriptors. Therefore, it is sufficient to rotate a tonal vector of the right number of semitones (i.e. at most 11 shifts) to achieve a similar representation for the audio excerpts, despite the original tonal difference.

Tonal descriptors have been applied in different MIR scenarios, such as chord recognition [39], cover identification [15, 35, 69, 85], and audio thumbnailing [7]. We refer the reader to [24] and [41] for further details and literature about tonality analysis.

2.2.5 Audio Fingerprinting

The requirement of a compact and efficient representation of the music content has been the main driver for the development of *audio fingerprinting* techniques. An audio fingerprint is a unique set of music features extracted from the audio signal that allows for the identification of digital copies of a music item even in presence of additional noise, distortion, D/A and A/D conversion, and lossy compression. It can be seen as a content-based signature that summarizes an audio recording. A comprehensive introduction to audio fingerprinting has been provided by Cano et al. [16]. A number of commercial applications are available; an example is Shazam![5] that, given about ten seconds of an unknown recording, provide users with meta-data and directions on where to purchase it (if that unknown recording is stored in the Shazam! collection). Additionally, fingerprinting algorithms are used to remove nearly duplicates in large collections, as it is carried out by Last.fm. Also copyright infringement detection makes use of fingerprinting as this task require finding differently encoded versions of the same piece of music. Given their commercial applications, audio fingerprinting techniques are designed to identify a particular performance of a given music work and they usually cannot – and perhaps are not meant to – identify different performances of the same music work (e.g. lives and covers).

2.2.6 Music Identification

The audio fingerprint is computed over the audio signal of a song for uniquely characterizing that specific recording; therefore, it is not able to generalize the features and to identify different performances of the same music work. On the other hand, the identification of a music work may be carried out also without linking the process to a particular recording. For example, music identification of broadcast live performances may not benefit from the fingerprint of other performances, because most of the acoustic parameters may be different. Additionally, in the case of classical music, the same works may have hundreds of different recordings, and it is not feasible to collect all of them in order to create a different fingerprint

[5] `http://www.shazam.com/`

of every one.

The goal of a music identification system is to identify a music work from the recording of a performance, yet independently from the particular performance. Ideally, one high-quality recording of a music work (i.e. the studio version) stored in a music collection can be used to identify all the other recordings of the same work (i.e. covers, lives, etc.). The approach can be considered a generalization of audio fingerprinting, because the relevant features used for identification are not linked to a particular performance of a music work. It should be noted that the approach allows the users to identify the metadata related to a musical work, but not to the particular performance used for the identification. Nevertheless, this limitation is balanced by the fact that only a song needs to be stored in the database.

The application scenario is as follows. A music excerpt of variable duration is submitted to the system, which provides as final result a list of music works ranked by their similarity with the query. The system should guarantee that the matching item is ranked in the first n positions, with n that ideally approach to one[6]. The final user is required to listen to the retrieved recordings for finding the correct item; the fact that the latter is expected at the beginning of the list allows the user to save time and efforts.

In literature, approaches for music identification have been proposed for classical music [28, 53, 68], as well as for popular music [8, 15, 85]; in this latter case, the task is also defined as *cover identification*. The next section overviews the system that we developed for classical music identification, which is likely the applicative scenario that mostly needs automatic strategies for cataloguing (analog) unknown content. This system was built within the FP6 European SAPIR project[7], the goal which was to develop a large-scale, distributed peer-to-peer architecture that makes it possible to search audio-visual content using the query-by-example paradigm. Further details about the system can be found in [66–69].

2.2.6.1 A Scalable System for Classical Music Identification

The structure of the system is depicted in Figure 2.1; there are two main components. The first carries out efficient and scalable retrieval for extracting a small subset of songs; the goal is to have the matching item in this subset. The second performs a statistical alignment between the query and each song in the subset and deliver the final ranking list.

(a) Indexing and Retrieval via Chroma Hashing

The task of retrieving a subset of candidates has to be performed efficiently to guarantee a high degree of scalability when in presence of large corpora. At this aim, *indexing* has always been an important research topic in modern information retrieval, and many method-

[6] We consider a scenario where the collection only stores at most one matching item for each query. However, the task could also be extended to collections that store several matching items for each query.

[7] SAPIR is the acronym of Search In Audio Visual Content Using Peer-to-peer Information Retrieval, a FP6 European STREP (Specific Targeted REsearch Project) last 30 months, since January 2007 to June 2009 (`http://www.sapir.eu/`).

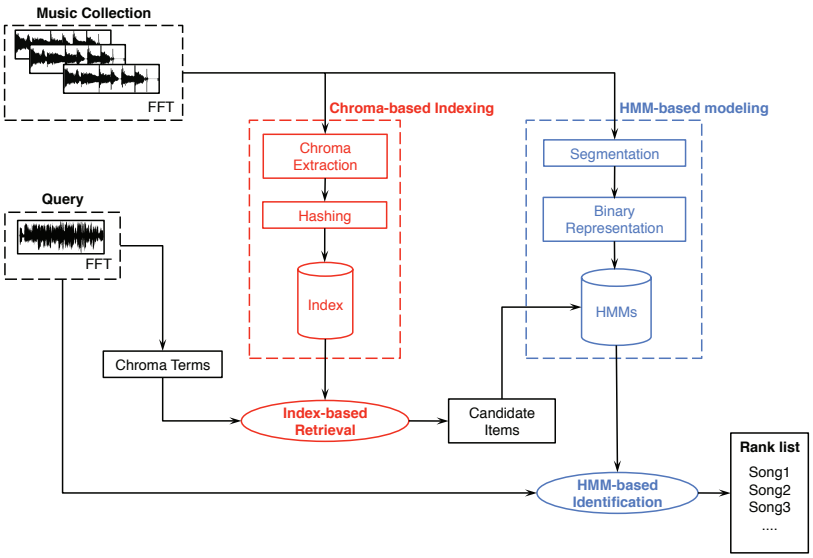


Figure 2.1. General structure of the music identification system.

ologies have been developed for handling collections of textual documents [4, 62]. A major advantage of indexing is that the list of index terms can be accessed in logarithmic, or even constant, time. The same cannot apply to the multi-dimensional features typical of music representation, because exact match has to be replaced by similarity search, which is less efficient.

With the aim of scalability we propose to use audio features as index terms, that is pointers to the recording they belong to, playing the same role of words in textual domain. In particular, we apply an hashing function to a quantized representation of 12-dimensional chroma vectors (see Section 2.2.4) for summarizing each vector as an integer [69]. Therefore, each song results described by a sequence of integers (i.e. *chroma words*), rather than a sequence of multi-dimensional vectors. See Appendix A for the details about the computation of this word-like representation. The initial idea stems from the concept of Locality Sensitive Hashing (LSH) [40], which has been applied to efficient search of different media [89]. LSH is a general approach to handle high dimensional spaces by using ad-hoc hashing functions to create collisions between vectors that are close in the high dimensional space. The advantage of hashing the music descriptors against general LSH is that the resulting words do not depend on the statistical properties of a particular collection and, therefore can be shared between different collections.

At indexing time, all the chroma words describing each song of a collection are added to a data structure based on a balanced binary search tree, where search operations can be performed in logarithmic time on the number of terms [26]. Additionally, timing information for each index term can be added to the data structure to store their position along the songs. The index can be post-analysed to remove the *stop words*, which are very common or very

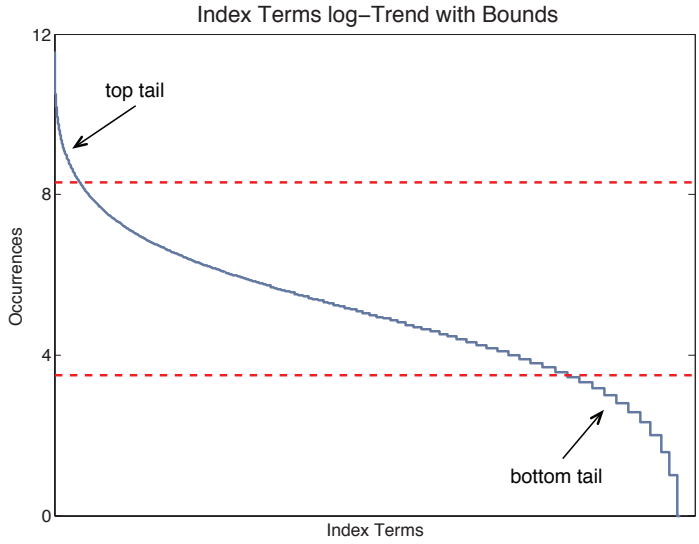


Figure 2.2. Distribution of the index terms sorted by their occurrence (logarithmic form) with highlighted potential bounds for stop words removal; the index refers to a collection of about 13,000 classical music recordings.

rare terms that could decrease the effectiveness of the retrieval process[8]. It is expected that removing stop words would improve effectiveness, while having at the same time a positive impact on scalability because the number of elements stored in the binary search tree may reduce significantly. The presence of stop words in a music index is highlighted in Figure 2.2, which depicts the distribution of the chroma words sorted by their occurrence (logarithmic form) in an index built over a collection of about 13,000 classical music recordings. The presence of stop words lead to top and bottom tails which can be reduced by (upper and lower) bounding the number of admitted term occurrences. Potential bounds are reported in Figure 2.2 as well.

Retrieval is carried out using the straightforward *bag-of-words* paradigm, then just by counting the number of common words between the query and the songs of the collection. As of Section 2.2.4, a chroma representation is sensitive to tonality shifts. This limitation is addressed through a mechanism of query expansion, where each query is represented by 12 vectors, that is one vector for each semitone (i.e. the original version and the eleven possible shifts). See the final part of Appendix A for the details about how to apply tonality shift to chroma words. At the end of the retrieval process, the set of songs with the largest number of words in common with the query are included in the subset of candidates. The proposed similarity value is not very common in textual information retrieval, where more complex measures are generally exploited, such as the cosine of the angle between the vectors representing the query and the items in the database, or the $tf \cdot idf$. However, we discov-

[8]In the textual information retrieval, the term *stop words* is synonym of very common words only. However, in order to simplify the description, we use *stop words* to denote both rare and common words.

ered empirically that the simple bag-of-words approach outperforms these more sophisticated measures.

(b) Statistical Modeling and Identification

Each song included in the subset of candidates is aligned with the audio query through an application of hidden Markov models (see Section 3.3) in order to increase the precision of the system [68]. In this part, the audio content is represented by the binary descriptors proposed in Appendix B. Briefly, each song is divided in segments that group consecutive audio frames with a coherent acoustic content [9]. Each segment is then summarized by a set of acoustic parameters related to the pitch of the audio signal (i.e. the presence of peaks in the frequency representation of an audio frame) that are general enough to be shared by different recordings of the same music work.

The binary descriptors are used to build a HMM that represents a song as a double stochastic process. In particular, the basic idea is that the states of the model are labeled with events in the audio recording (i.e. their number is proportional to the number of segments), transitions model the temporal evolution of the audio recording and observations are related to the binary descriptors which describe a segment and help to distinguish different events. Each model is strictly *left-to-right*, in the sense that each state can perform only transitions towards subsequent states, with a positive impact on computational complexity. At identification time, all the models in the subset are ranked by the probability of having generated the acoustic features of the unknown recording. That is, the query is aligned with every model of the subset, and each model provides a measure about how likely it represents the audio content of the query; models are then ranked according to this likelihood value (i.e. solution to Problem 1 of Section 3.3.1). In addition, a correction factor is applied to the alignment in the form of a regression line, which measures if the alignment across the model can be considered feasible (i.e. not too slow or too fast, follows a linear trend, etc.). This regression factor can be used to discard models that output infeasible alignments [66].

(c) Experimental Results

The proposed system has been empirically evaluated using a collection of about $13,000$ recordings of classical music for a total of about 350 hours of music. In most cases, the music files constitute individual movements of musical works, whereas in fewer cases, they represent an entire work. The query set was composed of 50 recordings representing different performances of as many different music works stored in the collection. These recordings had a duration of about 2 minutes, and for each one we extracted two not overlapping random excerpts of 15 seconds; this led to 100 different queries. We used the Mean Reciprocal Rank (MRR) to evaluate the system; for each query MRR ranges between 0 and 1 and measures

[9] The aim of segmentation is to divide the music signal into sub-sequences that are bounded by the presence of music events, where an event occurs whenever the current pattern of a music piece is modified (i.e. one or more new notes being played or stopped).

at what level of the ranking list the matching item is placed. See Section 6.2.2 for a formal definition of MRR.

Experiments showed that the final system leads to efficient and effective results that improve the two components taken separately. The index-based retrieval alone achieved MRR = 0.64 with the index optimized by stop words removal ($MRR = 0.59$ before optimization). However, the most important point is that all the queries were correctly matched within the first 100 positions (i.e. the worst matching item was ranked as 67^{th}); besides, retrieval time was about 2 seconds per query on a single processor machine[10]. Conversely, the HMM-based identification alone achieved MRR = 0.71, but with some queries correctly matched out of the first 100 positions. Additionally, HMM-based identification was very slow (about 4 minutes per query); this depends by the fact that the alignment process has to be performed linearly over all the songs of the collection. When we combined the two components, using 100 candidates from the index-based retrieval, we achieved MRR = 0.85 with all the relevant songs correctly ranked in the first 20 positions for each query. This means that a user should generally listen to at most 20 songs for finding the matching item. Identification time is reduced considerably with respect to the HMM-based identification alone, being now about 4 seconds per query. Therefore, these results show how the system could be effectively used to label unknown recordings of classical music in a semi-supervised way, where also non-expert users can spend little time to catalog unknown recordings.

2.2.7 Automatic Tagging

The design of automatic music tagging systems (auto-taggers) has received a significant amount of attention in the last years [11]. The design of auto-taggers mainly focuses on predictive statistical models that capture the acoustic content of songs associated with a specific tag of a vocabulary. State-of-the-art auto-taggers are based on discriminative approaches – e.g. boosting [10] and support vector machines (SVMs) [61, 72] – as well as generative models - e.g. Gaussian mixture models (GMMs) [101], the codeword Bernoulli average (CBA) model [48], and the time series model [27]. Since auto-taggers are particularly relevant for the aim of this thesis, an extensive review of a selected number of approaches is given in Section 4.3. Based on these tag models, most auto-taggers generate a vector of tag weights when annotating a new song for music search and retrieval. This vector may be interpreted as a *semantic multinomial* (SMN), i.e. a distribution characterizing the relevance of each tag to a song. A song is annotated by selecting the top-ranked tags in its SMN; additionally, given a tag query, retrieval can be performed by ranking songs according to the tag's probability in their SMN. Figure 2.3 shows an example of semantic multinomial over all 149 tags with highlighted the 10 most probable tags.

[10] We used a single processor machine along all this set of experiments using general implementations not optimized for the machine architecture.

[11] As a consequence, in 2008, the "Audio Tag Classification" task was introduced in the MIREX campaign [32].

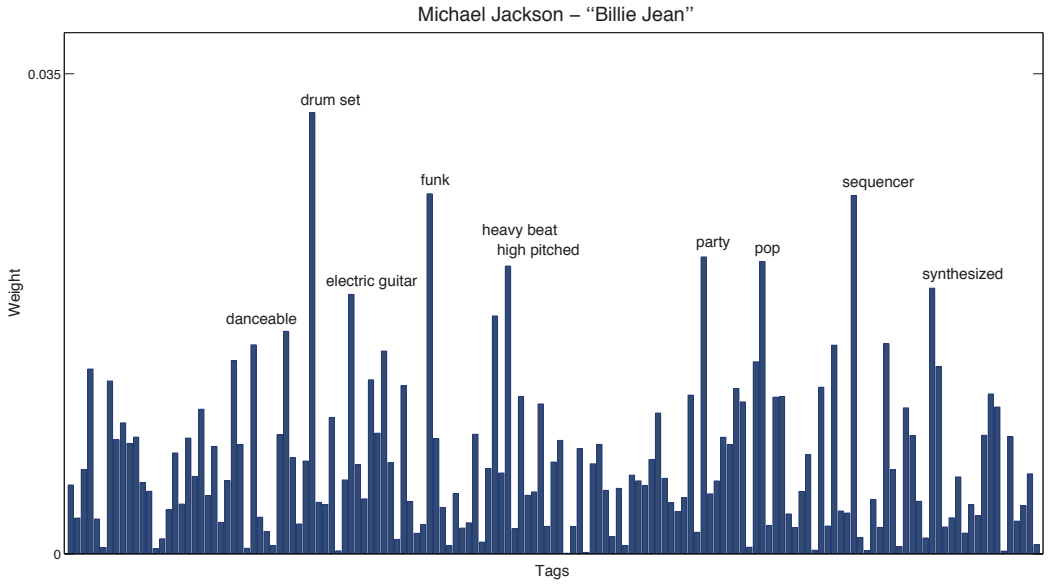


Figure 2.3. Semantic multinomial distribution over all 149 tags for the song: "Billie Jean", Michael Jackson; the ten most probable tags are labeled.

Auto-taggers rely on predictive models whose parameters need to be estimated using highly reliable ground truth. The latter generally consists of human-based annotations gathered in several ways, such as surveys [100], online games [57], or by mining commercial music recommendation services [93]. In particular, online games exploit the collaboration among players to infer reliable annotations that can be used for training purposes. Generally, two players may be given the same audio clip and asked to agree on a description for it. Otherwise, players may be provided with different songs, and asked to describe that song to each other. In any case, the degree of agreement (or disagreement) of the player answers are processed in order to infer human-based annotations as well as new tags that can be added to the semantic vocabulary.

2.2.8 Music Search and Discovery

In music retrieval we can identify two distinct user cases concerning the research of music: music *search* and music *discovery* [97]. Music search is useful when users know which songs, album or artist they want to find (e.g. a user that wants to purchase the new U2 album from an online store). Music search can be performed by metadata (i.e. song titles, album titles, artists, etc.) as well as by audio content; query-by-fingerprint and music identification are examples of this last scenario. Additionally, there has been an academic interest in developing music retrieval systems based on human performances, where the performances can be expressed by humming [29], singing [87], by playing on a MIDI device (usually a keyboard) [102], and so on. However, it may be difficult, especially, for untrained users, to emulate music dimensions well enough to make these systems effective [30].

In contrast, music *discovery* is a less direct pursuit in which users are not looking for a specific song or artist, but may have some general criteria that they wish to satisfy (e.g. "something rock", "something Beatles-like"). The personalized radios and recommendation systems introduced in Section 1.1.3 are examples of discovery systems. Discovery involves finding music previously unknown to the listener. Common discovery paradigms are query-by-popularity (e.g. browse the most downloaded songs), query-by-genre (e.g. search rock songs) and query-by-similarity (i.e. similarity with given songs or artists). In this last case, three types of similarity are mostly used to rank the songs: (i) acoustic similarity; (ii) social similarity (i.e. collaborative filtering), which finds music based on preference ratings or purchase sales records from a large group of users [21, 54]; (iii) semantic similarity, which consider common genre, instruments, vocal characteristics, and so on [9, 100]. More recently, a novel paradigm that is gaining increasing attention is *query-by-description* [100]. These queries are composed of tags that represent the information need; an example of such query is "mellow songs with slow tempo". All these paradigms can also be combined to build hybrid queries, such as "happy acoustic Beach Boys-like songs".

In the following of this thesis, we mostly focus on query-by-description and query-by-similarity (i.e. song similarity) paradigms. However, it should be noted that a complete approach for music search and discovery involves many (or all) of these paradigms. At present-day Last.fm offers several of them, including queries by metadata (artist, song, album, record label), by genre, by social similarity and by description (i.e. 1-tag and beta 2-tag queries)[12].

[12]It has to be noted that at the moment of writing this dissertation we did not have free access to many commercial discovery systems; we do not exclude that other services may satisfy more retrieval paradigms.

CHAPTER 3

Thesis Background

This chapter reviews some techniques and mathematical tools related to the work presented in the thesis. In particular, for the sake of completeness, we aim at providing the reader with additional formal notions and theories that may be useful to integrate the description of the methodologies reported in the following chapters. This is done to make the thesis as self-consistent as possible and to help a reader not particularly familiar with these concepts to better understand the work.

The following sections cover two different areas. In the first part, Section 3.1 reviews the algorithms used to represent the timbral content of the songs in the experiments reported in Chapter 6. In contrast, the second part reviews some statistical models and mathematical tools that can be considered the starting points for the original research carried out and presented in this thesis. In particular, Section 3.2 presents the Dirichlet distribution and the mixture models, while Section 3.3 introduce the hidden Markov models.

3.1 How to Represent the Timbre of Songs

The human perception of music similarity is subjective and context dependent. Relevant aspects of similarity include: instrumentation, timbre, melody, harmony, rhythm, tempo, mood, lyrics, sociocultural background, structure, and complexity. In order to compute effective content-based similarities it is necessary to extract the more appropriate features from the audio signal.

In this work, the applicative scenario is generally towards music retrieval systems where songs are ranked according to some sort of similarity relationships with a given query. Therefore, the music descriptors have to capture aspects in the audio content which help to understand whether songs are objectively similar, or not. For this reason, we mostly use one

of the music dimensions introduced in Section 2.1: the *timbre*. The latter is defined as the sound characteristics that allow listeners to perceive as different two sounds with same pitch and intensity, but played by different instruments. It is a complex notion also referred to as sound color, texture, or tone quality, and is derived from the spectral shape of an audio segment, independently of pitch and loudness.

We collect the songs timbre using different algorithms according to the way we could access a music corpus. In fact, in the MIR community music datasets can be generally released either completely as a set of audio clips (i.e. MP3 files) or just as a list of song details (i.e. title, artist, or other distinguishing metadata) for tackling copyright issues. These two scenarios and the corresponding descriptors are presented in the following sections. The reader should note that we will also integrate the timbre descriptors with other music dimensions whenever possible (i.e. when the audio clips are accessible). However, we prefer to report the details of these integrating features only when they are introduced in the work and focus this section on timbre only.

3.1.1 The Mel Frequency Cepstral Coefficients

When the audio clips are accessible, the timbre of the audio signal is widely represented by vectors of Mel Frequency Cepstral Coefficients (MFCC). MFCCs are a standard preprocessing technique in speech processing, which have proven to be useful for music information retrieval as well [79].

MFCCs summarize the spectral content of a short-time window (i.e. 20-50 ms) of an acoustic waveform by using the discrete cosine transform (DCT) to decorrelate the bins of a Mel-frequency spectral histogram. In particular, some very simple filters and transformations are applied to the audio signal for roughly modeling some of the characteristics of the human auditory system. These characteristics are: (i) the non-linear frequency resolution, (ii) the non-linear perception of loudness, and to some extent (iii) spectral masking effects[1].

The algorithms to extract MFCCs generally rely on the following steps [74]:

1. the audio signal is transformed from the time-domain to the frequency-domain since the human auditory system applies a similar transformation (in the *cochlea*, i.e. a part of the inner ear, different nerves respond to different frequencies). The audio signal is divided into short overlapping segments (e.g. 23 ms and 50% overlap) and a window function (e.g. Hann window) is applied to each segment; this is necessary to reduce spectral leakage. Third, the power spectrum matrix is computed using a Fast Fourier Transformation (FFT);

2. the power spectrum is mapped onto the Mel-scale using a filter bank consisting of triangular filters [92]. The Mel-scale is approximately linear for low frequencies (below 500 Hz), and logarithmic for higher frequencies. This scale has perceptual origins, since

[1] A tone is spectrally masked if it becomes inaudible by a simultaneous and louder tone with a different frequency.

it is used to better model the not-linear way humans perceive sound frequencies. Each triangular filter defines the response of one frequency band and is normalized such that the sum of weights for each triangle is the same. In particular, the height of each triangle is $2/d$ where d is the width of the frequency band. Adjacent triangles overlap each other so that the center frequency of one triangle is the starting point for the next triangle, and the end point of the previous triangle;

3. the logarithm of the powers is computed at each of the Mel frequencies (decibel). This is done to approximate the non-linear perception of loudness typical of the human auditory system;

4. the DCT is applied to compress the Mel power spectrum. In particular, the Mel frequency bands (e.g. 36) are represented by a list of coefficients (e.g. generally between 13 and 20). A side effect of the compression is that the spectrum is smoothed along the frequency axis; this can be interpreted as the simple approximation of the spectral masking in the human auditory system.

The MFCCs are the amplitudes of the resulting spectrum; they represent, in a very uncorrelated way, the information in the spectrum. There can be variations on this process, for example differences in the shape or spacing of the windows used to map the scale. For more details and implementations see [59, 74, 79]. In addition, we generally append to each MFCC vector its first and second instantaneous derivatives, achieving a final representation named delta-MFCCs. It has been empirically shown that this representation generally provides better performances in MIR tasks related to similarity [101].

3.1.2 The Echo Nest Timbre Features

When copyright issues prevent the owner from releasing the audio clips, a dataset can still be published as a list of songs specifying title, artist, and unique identification parameters (ID) that can be used for content access. In this scenario, descriptors can be obtained using public Web services, such as The Echo Nest Web Service. The Echo Nest is a music technology startup company that was founded in 2005 by Whitman and Jehan, two active MIR researchers from the MIT Media Lab. In addition to other services, The Echo Nest provides a free music analysis API[2] that can be used to collect information about songs. In particular, a user can upload an audio clip for analysis; once the track has been uploaded and automatically processed, the Echo Nest provides the user with a unique song ID that can be used to access the information about that song on the Echo Nest servers (i.e. content-based features as well as metadata). Releasing the unique song IDs makes it possible for everyone to access the same audio descriptions of a certain song. The content-based features available through The Echo Nest APIs capture different aspects of the music content, such as key,

[2] `http://developer.echonest.com/`

tempo, time signature, modality, average loudness, and so on; in particular, the Echo Nest Timbre (ENT) features provide the timbral descriptors.

ENTs are derived from slightly longer windows than MFCCs (generally between 100 and 500 ms), where, for each window, the Echo Nest service calculates 12 "timbre" coefficients. These coefficients highly abstract the spectral surface of the audio signal and are sorted by degree of importance. For instance, the first dimension represents the average loudness of the segment, the second emphasizes brightness, the third is more closely correlated with the flatness of a sound, and so on; nevertheless, the details of the exact calculation of these 12 coefficients is a trade secret of the company. At the end, a song is then represented by a bag of 12-dimensional vectors; computing the first and second instantaneous derivatives of these vector leads to the final 36-dimensional Delta-ENTs. ENT features have been successfully used in different applications related to music similarity and access (e.g. see [11, 93, 95]).

3.2 The Dirichlet Distribution and the Mixture Model

This section briefly overviews the Dirichlet distribution and the mixture models theory that we used to model the tag co-occurrences for improving state-of-the-art auto-taggers as described in Chapter 4. To the best of our knowledge, the latter is the first application of the Dirichlet distribution theory to MIR-related tasks. We refer the reader to [12, 31, 64] for more details[3].

The Dirichlet distribution is a multi-parameter generalization of the Beta distribution and defines a distribution over distributions, i.e. the result of sampling a Dirichlet is a distribution on some discrete probability space. It is often used as a model for proportional data such as the mix of vocabulary words in a text document. Let $\boldsymbol{p} = \{p_1, ..., p_{|\boldsymbol{p}|}\}$ denote a random vector whose elements sum to 1, so that p_j represents the proportion of item j with $j \in \{1, ..., |\boldsymbol{p}|\}$. Under the Dirichlet model with parameter vector $\boldsymbol{\alpha}$, the probability density at $\boldsymbol{p}$ is

$$\begin{aligned} P(\boldsymbol{p}) \sim \mathrm{Dir}(\boldsymbol{p};\ \alpha_1, ..., \alpha_{|\boldsymbol{p}|}) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_j (p_j)^{\alpha_j - 1}\ , \\ p_j > 0\ , \qquad \sum_j p_j = 1\ , \end{aligned} \tag{3.1}$$

where $\Gamma(n) = (n-1)!$ (i.e. Gamma function). The parameters $\boldsymbol{\alpha}$ can be estimated from a training set of N proportions, that is $D = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_N\}$. The maximum-likelihood estimate of $\boldsymbol{\alpha}$ maximizes $P(D|\boldsymbol{\alpha}) = \prod_{i=1}^{N} P(\boldsymbol{p}_i|\boldsymbol{\alpha})$. The log-likelihood can be written as

$$\log P(D|\boldsymbol{\alpha}) = N \log \Gamma(\sum_j \alpha_j) - N \sum_j \log \Gamma(\alpha_j) + N \sum_j (\alpha_j - 1) \log \bar{p}_j\ , \tag{3.2}$$

where

$$\log \bar{p}_j = \frac{1}{N} \sum_{i=1}^{N} \log p_{ij}\ . \tag{3.3}$$

[3] In this section, we report the formal theory about the Dirichlet distribution as presented by Minka [64].

There is no closed-form solution for the maximum-likelihood estimation; however, the objective is convex in $\boldsymbol{\alpha}$ since Dirichlet is in the exponential family. This implies that the likelihood is unimodal and the maximum can be found by a simple search. The gradient of the log-likelihood with respect to one α_j (i.e. first derivative) is

$$\begin{gathered} g_j = \frac{d \log P(D|\boldsymbol{\alpha})}{d\alpha_j} = N\Psi(\textstyle\sum_j \alpha_j) - N\Psi(\alpha_j) + N \log \bar{p}_j \ , \\ \Psi(x) = \frac{d \log \Gamma(x)}{dx} \ . \end{gathered} \tag{3.4}$$

Ψ is known as the digamma function and is similar to the natural logarithm. As always with the exponential family, when the gradient is zero, the expected sufficient statistics are equal to the observed sufficient statistics. In this case, the expected sufficient statistics are

$$E[\log p_j] = \Psi(\alpha_j) - \Psi(\sum_j \alpha_j) \ , \tag{3.5}$$

and the observed sufficient statistics are $\log \bar{p}_j$.

A fixed-point iteration for maximizing the likelihood can be derived starting with an initial guess for $\boldsymbol{\alpha}$ and constructing a simple lower bound on the likelihood which is tight at $\boldsymbol{\alpha}$. The maximum of this bound is computed in closed-form and it becomes the new guess. Such an iteration is guaranteed to converge to a stationary point of the likelihood since it is the same principle of the expectation-maximization (EM) algorithm [31]; for the Dirichlet, the maximum is the only stationary point.

A first approach to finding a stationary point relies on a bound on $\Gamma(\sum_j \alpha_j)$ which leads to the following fixed-point iteration:

$$\Psi(\alpha_j^{new}) = \Psi(\sum_j \alpha_j^{old}) + \log \bar{p}_j \ . \tag{3.6}$$

However this algorithm requires inverting the function Ψ, process which may lead to inefficiency problems.

An alternative approach is the Newton iteration [12]. The second derivative, i.e. the Hessian matrix, of the log-likelihood follows as (wherein $\mathbf{1}$ denotes the vector of all ones and $i \in \{1, ..., |\boldsymbol{p}|\}$):

$$\begin{aligned} \frac{d \log P(D|\boldsymbol{\alpha})}{d\alpha_j^2} &= N\Psi'(\textstyle\sum_j \alpha_j) - N\Psi'(\alpha_j) \ , \\ \frac{d \log P(D|\boldsymbol{\alpha})}{d\alpha_j d\alpha_i} &= N\Psi'(\textstyle\sum_j \alpha_j) \qquad (j \neq i) \ . \end{aligned} \tag{3.7}$$

Ψ' is known as the trigamma function. The Hessian can be written in matrix form as

$$\begin{gathered} \boldsymbol{H} = \boldsymbol{Q} + \mathbf{1}\mathbf{1}^T z \ , \\ q_{jj} = -N\Psi'(\alpha_j) \ , \quad q_{ij} = 0 \quad (i \neq j) \ , \\ z = N\Psi'(\textstyle\sum_j \alpha_j) \ . \end{gathered} \tag{3.8}$$

One Newton step is therefore:

$$\begin{aligned} \alpha_j^{new} &= \alpha_j^{old} - (\boldsymbol{H}^{-1}\boldsymbol{g})_j \ , \\ \boldsymbol{H}^{-1} &= \boldsymbol{Q}^{-1} - \frac{\boldsymbol{Q}^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{Q}^{-1}}{1/z+\mathbf{1}^T\boldsymbol{Q}^{-1}\mathbf{1}} \ , \\ (\boldsymbol{H}^{-1}\boldsymbol{g})_j &= \frac{g_j-b}{q_{jj}} \ , \\ b &= \frac{\sum_i g_i/q_{ii}}{1/z+\sum_i 1/q_{ii}} \ . \end{aligned} \tag{3.9}$$

Unlikely some other Newton algorithms, this one does not require storing or inverting the Hessian matrix explicitly, leading to very efficient implementations. Initialization can be performed randomly or using an approximate maximum likelihood estimation that finds the density which match the moments of the data [64].

In the auto-tagging system presented in Chapter 4 we use a mixture of Dirichlet models in order to better combine the samples, which may come from different populations. A Dirichlet Mixture Model is a linear superimposition of Dirichlet components, aimed at providing a richer class of density models than the single Dirichlet distribution. A mixture of Dirichlet distributions is defined as:

$$P(\boldsymbol{p}) \sim \sum_{k=1}^{K} \boldsymbol{\beta}_k \ \mathrm{Dir}(\boldsymbol{p};\ \boldsymbol{\alpha}_k) \ , \tag{3.10}$$

where K is the number of mixture components and $\boldsymbol{\Omega} = \{\boldsymbol{\beta}_k, \boldsymbol{\alpha}_k\}$ the vector of model parameters, with $\boldsymbol{\alpha}_k$ the parameters for the k^{th} Dirichlet mixture component and $\boldsymbol{\beta_k}$ the corresponding components weight vector. Given the training set of data, the parameters $\boldsymbol{\Omega}$ of the model are generally estimated using the generalized expectation-maximization (GEM) for maximum likelihood estimation [31]. We refer the reader to Section 4.4.1 for the algorithm and details concerning the parameter estimation with the Dirichlet mixture model.

3.3 The Hidden Markov Models (HMMs)

In this section we review the hidden Markov models (HMM) theory, which is the main tool for the retrieval framework presented in Chapter 5. In addition, we used HMMs in the music identification system introduced in Section 2.2.6.1-b as well. We refer the reader to [12, 38, 80] for more details about the HMM theory.

Although introduced and studied in the late 1960s and early 1970s, the HMMs have become increasingly popular in the late 1980s, in particular for two strong reasons. First the models are very rich in mathematical structure and can form the theoretical basis for use in a wide range of applications. Second the models, when applied properly, work very well in practice for several important applications, such as speech recognition [80]. In MIR, HMMs have been extensively used for query-by-example [87], alignment [71], segmentation [81], automatic identification [67], and chord recognition [49].

A hidden Markov model is a doubly embedded stochastic process with an underlying stochastic process that is *not* observable (it is hidden), but can only be observed through

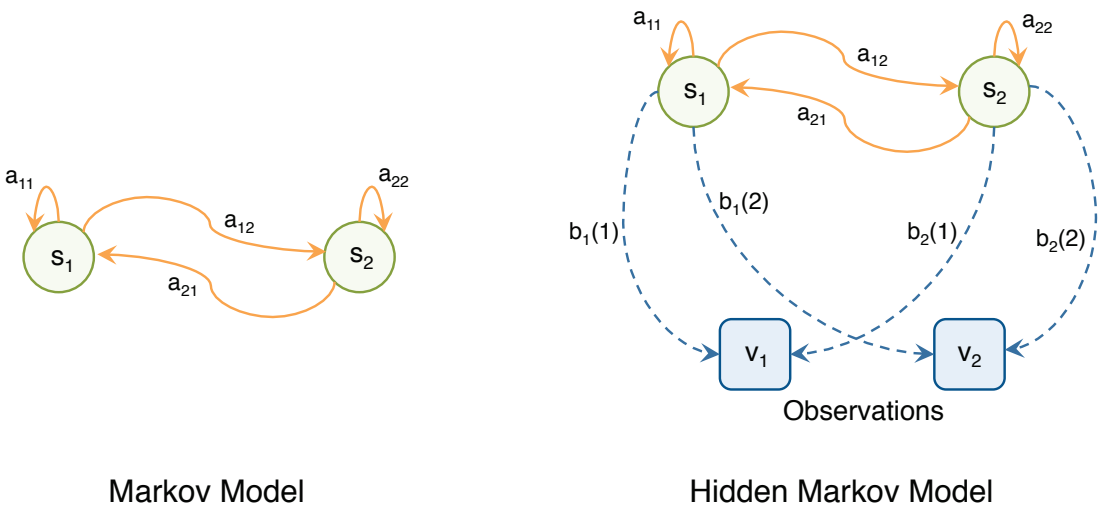


Figure 3.1. Comparison between observable and hidden Markov models (2 states): while in the observable model the outputs of the system are the states themselves, in the hidden models the outputs are ruled by a second stochastic process that produces a sequence of observations taken from the set $\{v_1, v_2\}$.

another set of stochastic processes that produce the sequence of observations. HMMs are an extension of observable Markov models (i.e. Markov chains) where the output of the process at each instant of time is the set of states, and each state corresponds to a physical (observable) event. Figure 3.1 highlights this main difference comparing observable and hidden Markov models; the models are composed of 2 states as well as 2 possible observations in the hidden case (the notation refers to the following paragraph). As can be seen, while in the observable models the output of a state is the state itself (i.e. the state is the physical event), in the hidden model the output of a state depends on a second stochastic distribution over a set of observations (i.e. a set of physical events); therefore, each state has a probability of being associated to an observations.

According to Rabiner [80], a HMM is characterized by the following:

1. N, the number of states in the model; although the states are hidden, they generally have some sort of physical significance. The states are denoted as $\boldsymbol{S} = \{S_1, ..., S_N\}$, and the state at time t as $s(t)$;

2. M, the number of distinct observation symbols per state, i.e. the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. Individual symbols are denoted as $V = \{v_1, ..., v_M\}$; note that the alphabet can be composed of continuous observation symbols as well;

3. the state transition probability distribution $\boldsymbol{A} = \{a_{ij}\}$ with

$$a_{ij} = P(s(t+1) = S_j \mid s(t) = S_i\} \ , \qquad 1 \leq i, j \leq N \ , \ \forall t \ . \tag{3.11}$$

For the special case where any state can reach any other state in a single step (i.e. fully connected model), we have $a_{ij} > 0$ for all i, j. For other HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs;

4. the emission symbol probability distribution in state j, $\boldsymbol{B} = \{b_j(m)\}$, where $1 \leq j \leq N$ and

$$b_j(m) = P(v_m \text{ at } t \mid s(t) = S_j) \ , \qquad 1 \leq m \leq M \ , \ \forall t \ . \tag{3.12}$$

5. the initial state distribution $\boldsymbol{\pi} = \{\pi_i\}$ where

$$\pi_i = P(s(1) = S_i) \ , \qquad 1 \leq i \leq N \ . \tag{3.13}$$

A HMM is usually specified with the notation $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ (N and M are inferred from $\boldsymbol{A}$ and $\boldsymbol{B}$). The HMMs can be used as both a generator of observations (i.e. $\boldsymbol{O} = \{o_1, o_2, ..., o_T\}$ where each observation o_t is one of the symbols from V, and T is the number of observations in the sequence), and as a model for how a given observation sequence was generated by an appropriate HMM.

3.3.1 The Three Basic Problems for HMMs

Considering the form of HMM of the previous section, there are three basic problems of interest that must be solved for the model to be useful in real-world application. In particular, given an observation sequence $\boldsymbol{O} = \{o_1, o_2, ..., o_T\}$ and a model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, these problems are the following:

Problem 1: how do we efficiently compute $P(\boldsymbol{O}|\lambda)$, the probability that the observed sequence was produced by the model?

Problem 2: how do we choose a corresponding state sequence $\bar{\mathcal{S}} = \{s(1)^*, s(2)^*, ..., s(T)^*\}$ which is optimal in some meaningful sense?

Problem 3: how do we adjust the model parameters $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ to maximize $P(\boldsymbol{O}|\lambda)$?

The next section describes the solution to Problem 2, which is the application involved with the retrieval framework presented in Chapter 5. In addition, the music identification system introduced ins Section 2.2.6.1-(b) concerns the solution of Problem 1 [66, 67]. In contrast, we never investigated solution of Problem 3 in MIR-related tasks.

3.3.2 The Viterbi Algorithm to Decode HMMs

Problem 2 attempts to uncover the hidden part of the model, that is to find the *correct* state sequence associated with the given observation sequence. However, there is no one single "correct" state sequence, since the correctness of a sequence is a related to some subjective optimality criteria.

One possible optimality criterion is to choose the states which are *individually* most likely (i.e. choose the most likely state $s(t)$ at each instant t without regard to the probability of occurrence of sequences of states). This approach maximizes the expected number of correct individual states; however it may lead to a not valid state sequence because of its intrinsic locality. For instance, when the HMM has state transitions $a_{ij} = 0$ for any pair (i, j), the resulting path could pass across infeasible edges. Alternatively, we can find the state sequence that maximizes the expected number of correct pairs of states $(s(t), s(t+1))$, or triple of states $(s(t), s(t+1), s(t+2))$, and so on.

Although all of these criteria might be reasonable for some applications, the most widely used approach is to find the globally optimal sequence across the model. A formal technique is based on a dynamic programming approach, and is called the Viterbi algorithm. The latter was conceived by Andrew Viterbi in 1967 as a decoding algorithm for convolutional codes over noisy digital communication links [38]. In the HMMs scenario it is applied to find the single best state sequence $\bar{\mathcal{S}} = \{s(1)^*, ..., s(T)^*\}$ for the given observation sequence $\boldsymbol{O} = \{o_1, ..., o_T\}$, i.e. the general idea is to maximize $P(\bar{\mathcal{S}}|\boldsymbol{O}, \lambda)$.

First, in order to describe the algorithm, we need to define the quantity:

$$\delta_t(i) = \max_{s(1),...,s(t-1)} P(s(1), ..., s(t-1), s(t) = S_i,\ o_1, ..., o_T \mid \lambda) \ , \tag{3.14}$$

that is the highest probability along a single path at time t, which accounts for the first t observations and ends in state S_i $(1 \leq i \leq N)$. By induction we have:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) \cdot a_{ij}] \cdot b_j(o_{t+1}) \ , \quad 1 \leq j \leq N \ . \tag{3.15}$$

In order to retrieve the state sequence, we need to keep track of the argument which maximized Equation 3.15, for each t and j; we do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can be stated as follows:

1. **Initialization:**

$$\begin{gathered} \delta_1(i) = \pi_i \cdot b_i(o_1) \ , \qquad \psi_1(i) = 0, \\ 1 \leq i \leq N \ . \end{gathered} \tag{3.16}$$

2. **Recursion:**

$$\begin{gathered} \delta_t(j) = b_j(o_t) \cdot [\max_i(\delta_{t-1}(i) \cdot a_{ij})] \ , \\ \psi_t(j) = \arg\max_i(\delta_{t-1}(i) \cdot a_{ij}) \ , \\ 2 \leq t \leq T \ , \quad 1 \leq i, j \leq N \ . \end{gathered} \tag{3.17}$$

3. **Termination:**

$$\begin{gathered} s(T)^* = \arg\max_i \delta_T(i) \ , \\ 1 \leq i \leq N \ . \end{gathered} \tag{3.18}$$

4. **State Sequence Backtracking:**

$$\begin{aligned} s(t)^* &= \psi_{t+1}(s(t+1)^*) \ , \\ t &= T-1,\ T-2,\ ...,\ 1 \ . \end{aligned} \tag{3.19}$$

The Viterbi algorithm can have a very efficient implementation related to the *forward* computation which is generally used to solve Problem 1; however this part is not relevant to the scope of this overview. We refer the reader to [80] for an extensive description about all the aspects around HMMs.

CHAPTER 4

Auto-tagging: Modeling Semantic Co-occurrences

The automated semantic annotation of musical content with descriptive tags is a core challenge in designing fully functional music retrieval systems [43]. In these systems, tags can be used for keyword search (e.g. searching for "mellow rock songs with acoustic guitar") or example-based retrieval founded on high-level semantic representations (e.g. generating playlists based on songs with similar annotations).

As already introduced in Section 2.2.7, to automatically annotate songs with semantic tags, based on audio content, auto-taggers model the characteristic acoustic patterns that are associated with each tag in a vocabulary. State-of-the-art auto-taggers rely on either discriminative or generative, where the resulting tag models are used to generate a vector of tag weights when annotating a new song. This vector represents a distribution characterizing the relevance of each tag to a song and is generally named *semantic multinomial* (SMN), as depicted in Figure 4.3, on the left hand side. The SMNs describing a song can then be used for music annotation and retrieval.

While some semantic associations in music are inspired by direct auditory cues (e.g. hearing a "violin"), others are inferred through contextual relationships (e.g. inferring "cello" and "bassoon", when listening to "orchestral classical music", or "high energy" when listening to "fast tempo"). These contextual relationships *correlate* tags. This is illustrated in Figures 4.1 and 4.2 and Table 4.1.

Figure 4.1 highlights tag co-occurrence patterns (e.g. "rock" songs also tagged as "guitar") in the CAL500 dataset [100], one of the annotated music collections used as *training* data in later experiments (see Section 6.1.1 for more details). The pairwise correlation between two tags is computed as the Jaccard coefficient [62], which measures the number of times both tags co-occur, over all songs, normalized by the total number of times the two tags appear. In particular, for n_{ij} the number of times tags w_i and w_j co-occur over all songs in the dataset,

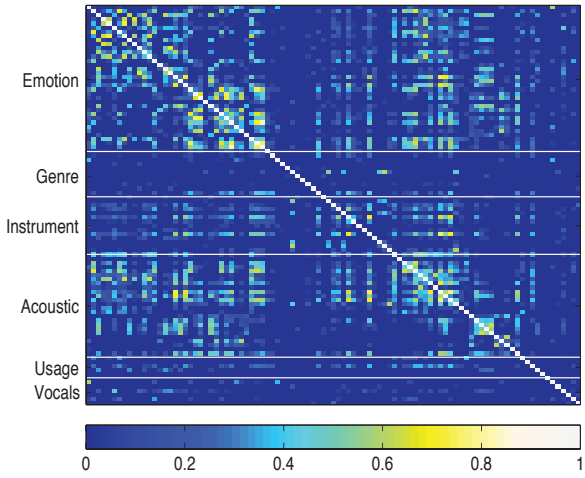


Figure 4.1. Tag co-occurrence in the CAL500 data set. Each row and column corresponds to exactly one tag of the CAL500 vocabulary, ordered by tag category. Entries indicate correlation between the corresponding tags, computed as the Jaccard coefficients, with values ranging from blue (mutually exclusive tags) to white (perfectly correlated tags). Emotion, instrument and acoustic tags exhibit significant co-occurrence with other tags.

Tag	Top-5 co-occurring tags
hard rock	angry, aggressive vocals, unpleasant, negative feelings, male lead vocals
acoustic guitar	acoustic, not exciting, folk, mellow, light beat
happy emotion	festive, positive feelings, optimistic, carefree, catchy
very danceable song	fast tempo, using at a party, cheerful, awakening, happy
going to sleep	calming, tender, mellow, slow beat, low energy

Table 4.1. Top-5 co-occurring tags for a sample of CAL500 tags.

the Jaccard coefficient c_{ij} is defined as $c_{ij} = n_{ij}/(n_i + n_j - n_{ij})$, where n_i represents the number of songs annotated with the tag w_i. The Jaccard coefficients range between 0 and 1 and are strictly positive if the tags are not mutually exclusive. As Figure 4.1 indicates, tags in the "Emotion", "Instrument" and "Acoustic" categories correlate significantly with other tags.

Additionally, Table 4.1 shows the top five co-occurrences for a sample of tags. Figure 4.2, on the other hand, shows that the SMNs (generated by a GMM-based auto-tagger) of three CAL500 songs that are associated with the tag "high energy" exhibit similar tag co-occurrences. While the auto-tagger predicts the relevance of the tag "high energy" to be small, based on each song's audio content, this could be corrected by leveraging contextual evidence provided, for example, by the tags "alternative" and "hard rock", which often co-occur in the SMNs of "high energy" songs.

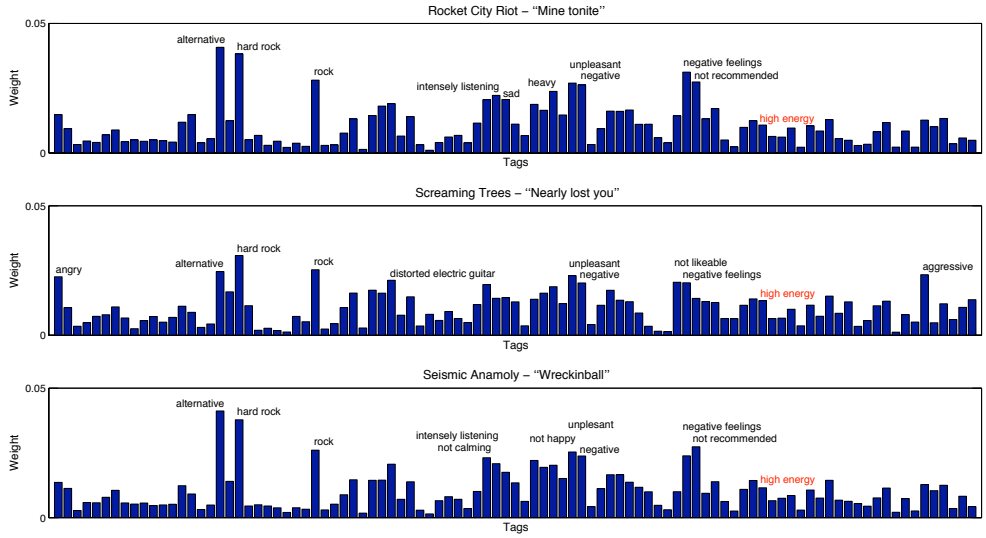


Figure 4.2. Examples of semantic multinomials of three songs associated with the tag "high energy" in the CAL500 data set. While the content-based auto-tagger fails at predicting the relevance of the tag "high energy", a context model could learn to correct this by observing contextual evidence, e.g., the clear co-occurrence of the tags "alternative" and "hard rock" in "high energy" songs.

Auto-tagging systems are expected to benefit from recognizing this context and the tag correlations it induces. For example, the prediction of a tag may be facilitated by the presence or absence of other tags — if a song has been tagged with "drums", the tag "electric guitar" is significantly more likely than "violin". Most content-based state-of-the-art auto-taggers, however, model each semantic tag *independently* and thereby ignore contextual tag correlations.

In this chapter, we introduce an additional layer of semantic modeling that supplements existing auto-tagging models by explicitly capturing tag correlations in SMNs. This is shown in Figure 4.3. Each tag in the vocabulary is considered to define a broader context that causes multiple, related tags to co-occur in a song's SMN. For each tag, we capture this broader context by estimating a *generative* context model, more specifically, a Dirichlet mixture model (DMM) of the tag distribution in the SMNs predicted for songs associated with the tag. To annotate a new, unlabeled song, the DMMs are used to post process the auto-tagger's SMN, based on semantic context. In particular, a DMM tag model will adjust (i.e. reduce or boost) a tag's SMN weight given contextual evidence provided by other co-occurring tags. This will allow us to correct auto-tagger errors in the SMNs, while reinforcing good posterior probability estimates. The resulting tag weights define a *contextual* multinomial (see Figure 4.3).

The proposed approach is flexible in that it can be combined with *any* auto-tagger that generates SMNs. As a *generative* model, it also handles weakly-labeled data [1] well: unlike

[1] In weakly-labeled data, the presence of a tag implies that it applies to a song; the absence of a tag,

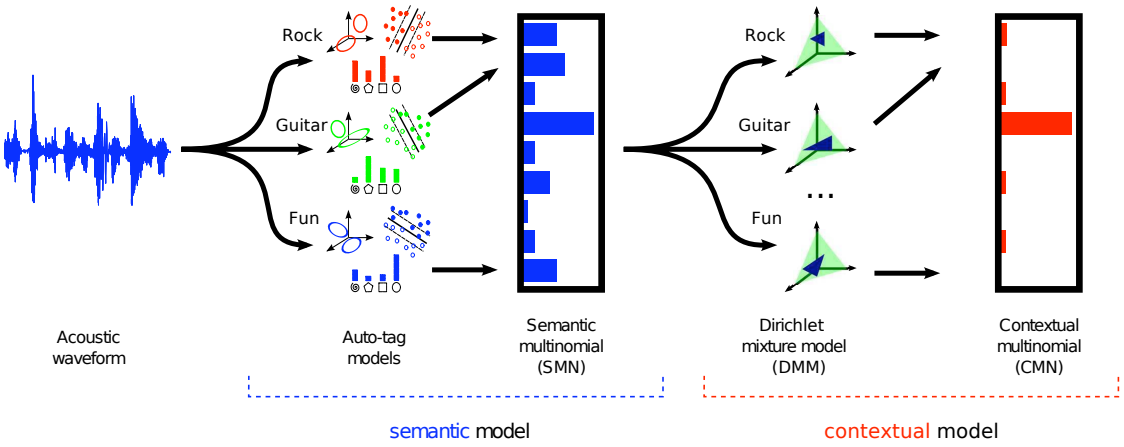


Figure 4.3. Overview of the system: DMMs model context by considering co-occurrence patterns between tags in the semantic multinomials.

discriminative models (e.g. SVMs, boosting, decision trees), which also require reliable negative training examples, generative models only require training examples that have been positively associated with a semantic tag, to estimate class-conditional distributions. Moreover, generative models are better suited at estimating distributions that naturally emerge around relevant contextual tag correlations in the SMNs, while down-weighting irrelevant outliers (e.g. accidental tag co-occurrences), and ranking tags probabilistically for a song (by applying Bayes' rule). Finally, as the Dirichlet distribution is the conjugate prior of the multinomial distribution, the DMM is a natural choice for modeling the distribution of SMNs generated by an auto-tagger.

The remainder of this chapter is organized as follows. After a discussion of related work in Section 4.1, Section 4.2 describes the music annotation and retrieval problem. In Section 4.3, we provide an overview of various auto-tagging systems that generate SMNs. Lastly, Section 4.4 introduces the DMM as a model to capture contextual tag correlations in semantic multinomials.

4.1 Related Work

The prohibitive cost of manual labeling of multimedia content (e.g. images, music, movies, etc.) made automatic annotation a major challenge in various fields of research. In the computer vision community, the automatic annotation and retrieval of images has been a topic of ongoing research (see, for example, [5,13,17,18,33,36,58,96,105]). Recently, Rasiwasia and Vasconcelos [83] proposed a framework that combines object-centric and scene-centric methods for modeling contextual relationships between visual concepts.

In music information retrieval, the design of auto-taggers has mainly focused on predictive statistical models that capture the acoustic content of songs associated with a specific tag

however, does not guarantee it does not apply.

(see, for example, [34, 48, 61, 75, 88, 101, 104, 107, 108]), where different tags are often modeled independently. When annotating a new, unlabeled song, most of these semantic tag models allow semantic multinomials to be computed. Some recent work has started to consider tag correlation [3, 10, 23, 72, 110]). Most of this work trains a *discriminative* context model for each tag, based on the tag weights (i.e. SMN) output by the semantic tag models. This context model is often very similar in nature to the semantic model it is "stacked" on top of. For example, Yang et al. [110] propose a discriminative approach based on ordinal regression for both the semantic and the context models (for each tag, training songs are assigned to 4 groups of decreasing relevance, based on empirically observed tag correlations in the ground truth annotations); Ness et al. [72] use support vector machines (SVMs) for semantic and context models, while Bertin-Mahieux et al. [10] apply two stages of boosting. Aucouturier et al. [3] use a decision tree to refine the result of individual detectors. More recently, Chen et al. [23] proposed estimating not only *word* models, but also *anti-word* models, using GMMs. Word models are regular semantic tag models which model the presence of a tag, as in the work of Turnbull et al. [101]. Anti-word models, on the other hand, capture acoustic content that is *not* usually associated with a word, by characterizing content that is associated with tags of opposite semantic meaning (i.e., tags that are negatively correlated with the word, on the training set). Predictions from both types of models are combined to obtain the final tag weights. While this approach refines the GMM-based semantic multinomials by explicitly accounting for acoustic patterns that may indicate a tag's absence, it does not provide a holistic contextual model for each tag, to leverage positive, negative and more complex tag correlation patterns in the SMNs.

In this work, we focus on developing a *generative* context model that can be combined with *any* existing auto-tagger that generates SMNs. Compared to the previously proposed discriminative approaches, a generative context model is expected to better handle weakly labeled data and has some other advantages, as discussed before (e.g. the model naturally provides probabilities to rank predictions). Since our context model is not being developed for a specific auto-tagger, we measure its benefit for a variety of auto-taggers by comparing their performance with and without being combined with the context model. We also explore how other approaches to capturing context (i.e. [10, 72, 110]) generalize in combination with auto-taggers other than the ones they were developed for.

4.2 Music Annotation and Retrieval

This section discusses the related tasks of annotation and retrieval of audio data as a supervised multi-class labeling (SML) problem ([18, 101]). In this setting, each tag in a vocabulary represents a class label, and each song is tagged with multiple labels. Many existing auto-tagging approaches fit within this framework. Section 4.3 describes several of them in more detail.

4.2.1 Problem Formulation

The acoustic content of a song $\mathcal{X}$ is represented as a bag of features, $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t$ is a vector of audio features extracted from a short snippet (e.g. 20-50 ms) of the audio signal, and T depends on the length of the song.

The semantic content of a song with respect to a vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$ is represented in an annotation vector of *semantic weights* $\mathbf{a} = [a_1, \ldots, a_{|\mathcal{V}|}]$, where $0 \leq a_i \leq 1$, with $a_i > 0$ only if the song has been positively associated with the i^{th} tag, w_i, otherwise $a_i = 0$. The actual value of each weight, a_i, represents the strength of association between the song and the tag. The data set $\mathcal{D} = \{(\mathcal{X}_d^{\boldsymbol{x}}, \mathbf{a}_d)\}_{d=1}^{|\mathcal{D}|}$ is a collection of $|\mathcal{D}|$ song-annotation pairs. Binary class labels can be obtained by mapping the semantic weights, a_i, to $\{0, 1\}$.

4.2.2 Annotation

We treat annotation as a supervised multi-class labeling problem, where each class corresponds to a tag w_i from the vocabulary $\mathcal{V}$ of $|\mathcal{V}|$ unique tags ("rock", "drum", "tender", etc.). Annotation involves finding the subset of classes, i.e., tags, that best describe an unseen song $\mathcal{X}$.

First, a set of models (classifiers, class-conditional densities, etc., depending on the type of auto-tagger) over the audio feature space is trained to recognize the acoustic content associated with each tag w_i in the vocabulary. In most auto-tagging systems, one model is trained per tag and models for different tags are trained *independently*. Then, based on these models, the posterior probability that each tag w_i applies to a song $\mathcal{X}$, i.e. $P(w_i|\mathcal{X}^{\boldsymbol{x}})$, is predicted. Finally, the song is represented as a semantic multinomial $\boldsymbol{\pi} = \{\pi_1, ..., \pi_{|V|}\}$ on the probability simplex (i.e. $\sum_i \pi_i = 1$ with $\pi_i \geq 0$), where $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ represents the relevance of the i^{th} tag for the song $\mathcal{X}$. A song is then annotated by selecting the tags with the largest posterior probability according to $\boldsymbol{\pi}$.

4.2.3 Retrieval

To retrieve songs given a tag-based query, all songs in a database are ranked based on their relevance to the query and the top-ranked results are returned to the user. More specifically, we determine the relevance of a song $\mathcal{X}$ to a query with tag w_i based on the posterior probability of the tag for that song, i.e. $P(w_i|\mathcal{X}^{\boldsymbol{x}})$. Hence, songs in the database are ranked based on the i^{th} entry, π_i, of their semantic multinomials $\boldsymbol{\pi}$. In this study, we focus on single-tag queries; yet this framework can be easily extended to multiple-tag queries [100].

4.3 Auto-Taggers to Compute Semantic Multinomials

We briefly review four state-of-the-art auto-tagging systems, which allow semantic multinomials to be computed as described in Section 4.2 and, therefore, may be leveraged with the contextual model presented in the previous section. More details about the models and

the audio features used by each of the reviewed auto-tagging algorithms can be found in the corresponding references. For our experiments, we generally used the parameter settings and the audio features reported in those earlier works[2]. However, when the audio clips were not accessible, we described the audio content using only the delta-ENT features introduced in Section 3.1.2 for all the auto-taggers.

4.3.1 Gaussian Mixture Models (GMMs)

This generative model was proposed by Turnbull et al. [101]. Each tag w_i, $i = 1, \ldots, |\mathcal{V}|$, in the vocabulary $\mathcal{V}$ is modeled with a probability distribution $P(\boldsymbol{x}|w_i)$ over the space of audio features $\boldsymbol{x}$, which is a Gaussian mixture model and captures the acoustic patterns that are associated with w_i:

$$P(\boldsymbol{x}|w_i) = \sum_{k=1}^{K} a_k^{w_i} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k^{w_i}, \boldsymbol{\Sigma}_k^{w_i}), \tag{4.1}$$

where K is the number of mixture components, $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and (diagonal) covariance matrix $\boldsymbol{\Sigma}$, and $a_k^{w_i}$ the mixing weights. For each tag model $P(\boldsymbol{x}|w_i)$, the parameters $\{a_k^{w_i}, \boldsymbol{\mu}_k^{w_i}, \boldsymbol{\Sigma}_k^{w_i}\}_{k=1}^{K}$ are *estimated* from the audio features of songs that are positively associated with w_i, using an efficient hierarchical expectation-maximization algorithm [101].

Given the audio content $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ of a new song $\mathcal{X}$, a Bayes decision rule based on the posterior tag probabilities achieves the minimum probability of error to infer relevant tags:

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = \frac{P(\mathcal{X}^{\boldsymbol{x}}|w_i)\, P(w_i)}{P(\mathcal{X}^{\boldsymbol{x}})}, \tag{4.2}$$

where $P(w_i)$ is the prior of the i^{th} tag (assumed to be uniform) and $P(\mathcal{X}^{\boldsymbol{x}})$ the song prior, computed as $P(\mathcal{X}^{\boldsymbol{x}}) = \sum_{j=1}^{|\mathcal{V}|} P(\mathcal{X}^{\boldsymbol{x}}|w_j)P(w_j)$. The likelihood term in Equation (4.2), $P(\mathcal{X}^{\boldsymbol{x}}|w_i)$, is estimated with the geometric average $\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_i)\right)^{1/T}$. Finally, we have:

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = \frac{\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_i)\right)^{\frac{1}{T}}}{\sum_{j=1}^{|\mathcal{V}|}\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_j)\right)^{\frac{1}{T}}}. \tag{4.3}$$

We run experiments using the code of Turnbull et al. [101], estimating song models with 8 and tag models with 16 mixture components. Songs are represented using timbral descriptors only; in particular, audio segments are summarized by 39-dimensional Delta-MFCC feature vectors (i.e. 13 MFCC components) computed by extracting half-overlapping, short-time windows (of 23 ms) from the audio signal.

[2]The description of each auto-taggers include the presentation of the audio features reported in the corresponding references that we used when the audio clips were accessible.

4.3.2 Codeword Bernoulli Average (CBA)

CBA [48] is a probabilistic model for predicting the probability that a tag applies to a song based on a vector-quantized representation of the audio features with M codewords. So, rather than representing a song as a bag of audio feature vectors $\boldsymbol{x}_t$, CBA represents a song $\mathcal{X}$ as a vector $\mathbf{n}_\mathcal{X}$ of counts of M codewords.

For a song $\mathcal{X}$, the CBA model generates a set of binary random variables $y_i \in \{0, 1\}$, $i = 1, ..., |\mathcal{V}|$, which determine whether or not each tag w_i applies to $\mathcal{X}$, in two steps. First, for each tag w_i, $i = 1, ..., |\mathcal{V}|$, a codeword $z_i \in \{1, ..., M\}$ is selected with probability proportional to the number of times it appears in the song, i.e. proportional to the corresponding entry of $\mathbf{n}_\mathcal{X}$. Then, y_i is sampled from a Bernoulli distribution with parameter $\beta_{z_i i}$ (denoting entry (z_i, i) of the parameter matrix $\boldsymbol{\beta}$):

$$\begin{aligned} P(y_i = 1|z_i, \boldsymbol{\beta}) &= \beta_{z_i i} \ , \\ P(y_i = 0|z_i, \boldsymbol{\beta}) &= 1 - \beta_{z_i i} \ . \end{aligned} \tag{4.4}$$

The Bernoulli parameters $\boldsymbol{\beta}$ are estimated from training data using the expectation-maximization algorithm [31].

The probability that a tag w_i applies to an unseen test song $\mathcal{X}$ can be inferred as

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = P(y_i = 1|\mathbf{n}_\mathcal{X}, \boldsymbol{\beta}) = \frac{1}{T} \sum_{m=1}^{M} (\mathbf{n}_\mathcal{X})_m \, \beta_{mi} \ , \tag{4.5}$$

where $\mathbf{n}_\mathcal{X}$ is the codebook representation of $\mathcal{X}$, and T its total number of features, with $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_t\}_{t=1}^{T}$.

We obtained the authors' code [48] to run our experiments. We modified it so the codebook is constructed using only songs from the training set and set the codebook size $M = 500$. Audio segments are represented as a bag of 39-dimensional Delta-MFCCs (same configuration of Section 4.3.1).

4.3.3 Boosting (BST)

The boosting approach of Eck et al. [34] is a discriminative approach that learns a binary classifier for each tag w_i in the vocabulary $\mathcal{V}$, from positive and negative training examples for that tag. More specifically, it constructs a *strong classifier* from a set of simpler classifiers, called *weak learners*, in an iterative way. As weak learners, we use single stumps (i.e. binary thresholding on one feature). Again, a test song $\mathcal{X}$ is classified by each of the binary classifiers and Platt scaling applied to produce a probability estimate $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ for each tag w_i. We obtained the authors' code [34] to run our experiments. The audio signal is described as a collection of 20-dimensional MFCCs, to capture timbral aspects, and a series of auto-correlation coefficients (computed for lags spanning from 250 ms to 2000 ms at 10 ms intervals), to describe tempo and pitch. Both sets of features are computed from 100 ms windows extracted from the audio signal every 75 ms.

4.3.4 Support Vector Machines (SVMs)

Mandel and Ellis use a collection of SVM classifiers for music auto-tagging [61]. An SVM is a discriminative, supervised learning algorithm that learns the maximum margin hyperplane separating two classes of data to construct a binary classifier. For a vocabulary with $|\mathcal{V}|$ tags, an SVM is learned for each tag w_i to predict the presence or absence of the tag. Each SVM is trained from a collection of positive and negative training examples for the tag it models, using a radial basis kernel.

To tag an unseen test song $\mathcal{X}$, it is classified by each of the $|\mathcal{V}|$ SVMs. Platt scaling [78] is used to convert distance from the separating hyperplane to a probability estimate $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ for each tag w_i.

We implement this model using LibSVM [22], setting the width of the radial basis kernel to 1 and the regularization parameter, C, to 10, after normalizing the data, as in the work of Mandel and Ellis [61]. The audio signal is described using timbral descriptors and short-term temporal features (to summarize beat, tempo, and rhythmic patterns). The timbral descriptors are obtained as the mean and unwrapped covariance of a clip's 18-dimensional MFCCs, computed from 25 ms windows, extracted every 10 ms. The temporal features are obtained by first combining the Mel frequency bands (of the same 25 ms windows) into low, low-mid, high-mid, and high frequencies, and then modeling the total magnitude in each of these four (large) frequency bands over time (by using a DCT to decorrelate the magnitudes of the Fourier transform of each band).

4.4 The Dirichlet Mixture as Generative Context Model

Instead of modeling tags independently, in this work we recognize contextual tag *correlations* and explicitly model them. We want the model to be (i) generally applicable, so it can be combined with *any* auto-tagger that generates SMNs, (ii) compatible with the SML framework proposed in Section 4.2 — i.e. generate tag multinomials for annotation and retrieval — , and (iii) generative (to handle weakly labeled data and estimate class-conditional distributions around meaningful tag co-occurrence patterns in SMNs, while down-weighting accidental co-occurrence patterns as outliers).

To achieve these objectives, each tag is considered as a "source" of a broader *context* that causes several related tags to co-occur in a song's SMN, and this context is modeled with a Dirichlet mixture model (DMM). Therefore, the DMM captures the typical co-occurrence patterns of tags in the SMNs for songs associated with the tag. As the conjugate prior of the multinomial distribution, the Dirichlet distribution is an appropriate model for this generative approach. The SMNs it models could be generated by any algorithm for automatic annotation. This gives rise to the two-level architecture depicted in Figure 4.3, applying DMM in tandem with any first-stage auto-tagger generating SMNs. After training the auto-tagger and estimating DMMs for all tags, a new, unlabeled song is tagged as follows. First, the auto-tagger of choice annotates the song with an SMN (left hand side of Figure 4.3). Then, in the

right half of Figure 4.3, each tag's SMN weight is adjusted based on contextual evidence, i.e. by evaluating the likelihood of co-occurring tags in the SMN based on the DMM for that tag.

This maps the song's *semantic* multinomial $\boldsymbol{\pi}$ into a *contextual* multinomial $\boldsymbol{\theta} = (\theta_1, ..., \theta_{|V|})$ (CMN), where $\theta_i = P(w_i|\boldsymbol{\pi})$ is the (context adjusted) relevance of the i^{th} tag, given the song's semantic multinomial $\boldsymbol{\pi}$. This new multinomial representation of songs can be used directly in the annotation and retrieval framework presented in Section 4.2, where each song is annotated with the most likely tags according to $\boldsymbol{\theta}$ (i.e. we select the tags showing the largest probability θ_i).

The remainder of this section explains in more detail how the DMMs are estimated (to model context) and then applied (leveraging context) to annotate new songs with a contextual multinomial.

4.4.1 Learning DMM Context Models for Each Tag

To model the context induced by a tag w_i, a Dirichlet mixture model is estimated based on the SMNs of all training songs in $\mathcal{D}$ associated with w_i. Note that to annotate a new song, as depicted in Figure 4.3, the DMM tag models are applied to the SMN *predicted* by some first-stage auto-tagger. To estimate the DMMs consistently, they are therefore estimated based on the SMNs predicted by the first-stage auto-tagger, for the songs associated with w_i. Moreover, to increase the size and diversity of the training data set, we will represent each song as a collection of SMNs by extracting (overlapping) 5-second segments from the song, every 3 seconds, and annotating each segment with a SMN. This results in a training set of N_{w_i} semantic multinomials, $\{\boldsymbol{\pi}^n\}_{n=1}^{N_{w_i}}$, associated with the tag w_i. Finally, to better emphasize the peaks in the training SMNs, we consider only the top-h tag weights in each SMN and decrease all others to very low values. In particular, for each tag w_i, we compute the kurtosis[3] k_n for each training SMN, $\boldsymbol{\pi}^n$, where $n = 1, ..., N_{w_i}$. For more peaked SMNs, with high kurtosis (i.e. $k_n > \bar{k}$), we select $h = 0.05 \cdot |\mathcal{V}|$ tags, while for more uniform SMNs (i.e. $k_n \leq \bar{k}$), we select $h = 0.1 \cdot |\mathcal{V}|$ tags. $\bar{k}$ is computed as the average kurtosis over all training SMNs associated with w_i, i.e. $\bar{k} = (\sum_{n=1}^{N_{w_i}} k_n)/N_{w_i}$. See Section 6.3.2.1 for details about this choice.

These SMNs are modeled as samples from a mixture of Dirichlet distributions [65],

$$P(\boldsymbol{\pi}|w_i; \Omega^{w_i}) = \sum_{k=1}^{K} \beta_k^{w_i} \ \mathrm{Dir}(\boldsymbol{\pi};\ \boldsymbol{\alpha}_k^{w_i}), \tag{4.9}$$

where K is the number of mixture components and $\Omega^{w_i} = \{\beta_k^{w_i}, \boldsymbol{\alpha}_k^{w_i}\}$ the vector of model parameters, with $\boldsymbol{\alpha}_k^{w_i}$ the parameters for the k^{th} Dirichlet mixture component and $\beta_k^{w_i}$ the corresponding component weight. The component weights are positive and normalized to sum

[3]Kurtosis provides a measure of the "peakedness" of the distribution, giving an indication whether a SMN is characterized by a few dominant peaks (high kurtosis value) or by a more uniform distribution (low kurtosis value).

Algorithm 1 GEM algorithm for DMM

1: **Input**: N semantic multinomials $\{(\pi_1^n, ..., \pi_{|V|}^n)\}_{n=1}^N$.
2: Randomly initialize DMM parameters $\Omega = \{\beta_k, \boldsymbol{\alpha}_k\}_{k=1}^K$.
3: Define

$$\Psi(x) = \frac{\mathrm{d}\log\Gamma(x)}{\mathrm{d}\,x} \quad \text{and} \quad \Psi'(x) = \frac{\mathrm{d}^2\log\Gamma(x)}{\mathrm{d}\,x^2} \ . \tag{4.6}$$

4: **repeat**
5: {E-step}
6: Compute responsibilities γ_k^n, for $k = 1, ..., K$ and $n = 1, ..., N$, based on current parameter values:

$$\gamma_k^n = \frac{\beta_k \cdot \mathrm{Dir}(\boldsymbol{\pi}^n; \boldsymbol{\alpha}_k)}{\sum_{j=1}^K \beta_j \cdot \mathrm{Dir}(\boldsymbol{\pi}^n; \boldsymbol{\alpha}_j)} \ . \tag{4.7}$$

7: {M-step}
8: Update DMM parameters. For $k = 1, ..., K$ (applying $\Psi, \Psi', /$ and log component-wise):

$$\begin{aligned} \boldsymbol{g}_k &= N\Psi(\mathbf{1}^T\boldsymbol{\alpha}_k)\mathbf{1} - N\Psi(\boldsymbol{\alpha}_k) + \textstyle\sum_n \log(\boldsymbol{\pi}^n) \ , \\ \boldsymbol{q}_k &= -N\Psi'(\boldsymbol{\alpha}_k) \ , \\ \boldsymbol{b}_k &= \frac{\mathbf{1}^T(\boldsymbol{g}_k/\boldsymbol{q}_k)}{\left(N\Psi'(\mathbf{1}^T\boldsymbol{\alpha}_k)\right)^{-1} + \mathbf{1}^T(\mathbf{1}/\boldsymbol{q}_k)} \cdot \mathbf{1} \ , \\ \beta_k^{\text{new}} &= \tfrac{1}{N}\textstyle\sum_{n=1}^N \gamma_k^n \ , \\ \boldsymbol{\alpha}_k^{\text{new}} &= \boldsymbol{\alpha}_k^{\text{old}} - \tfrac{\boldsymbol{g}_k - \boldsymbol{b}_k}{\boldsymbol{q}_k}. \end{aligned} \tag{4.8}$$

9: **until** convergence
10: **Output**: DMM parameters $\Omega = \{\beta_k, \boldsymbol{\alpha}_k\}_{k=1}^K$

to one, i.e. $\sum_k \beta_k^{w_i} = 1$ with $\beta_k^{w_i} \geq 0$. A Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha})$ with parameters $\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_{|V|}\}$ is given by:

$$\mathrm{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{|V|} \alpha_i)}{\prod_{i=1}^{|V|} \Gamma(\alpha_i)} \prod_{i=1}^{|V|} (\pi_i)^{\alpha_i - 1} \ , \tag{4.10}$$

where $\Gamma(\cdot)$ denotes the Gamma function.

Given the training set of SMNs associated with the tag w_i, i.e. $\{\boldsymbol{\pi}^n\}_{n=1}^{N_{w_i}}$, the parameters Ω^{w_i} of the contextual tag model are estimated by adopting the generalized expectation-maximization (GEM) algorithm for maximum likelihood estimation [31]. GEM is an extension of standard EM and can be applied when the maximization step (M-step) in standard EM is intractable. Just like EM, GEM is an iterative algorithm that iterates between an expectation step (E-step) and a (generalized) M-step. The E-step is identical to the E-step of standard EM: expectations are computed for each mixture component. The (generalized) M-step updates the estimates of the model parameters $\Omega^{w_i} = \{\beta_k^{w_i}, \boldsymbol{\alpha}_k^{w_i}\}$. Instead of solving

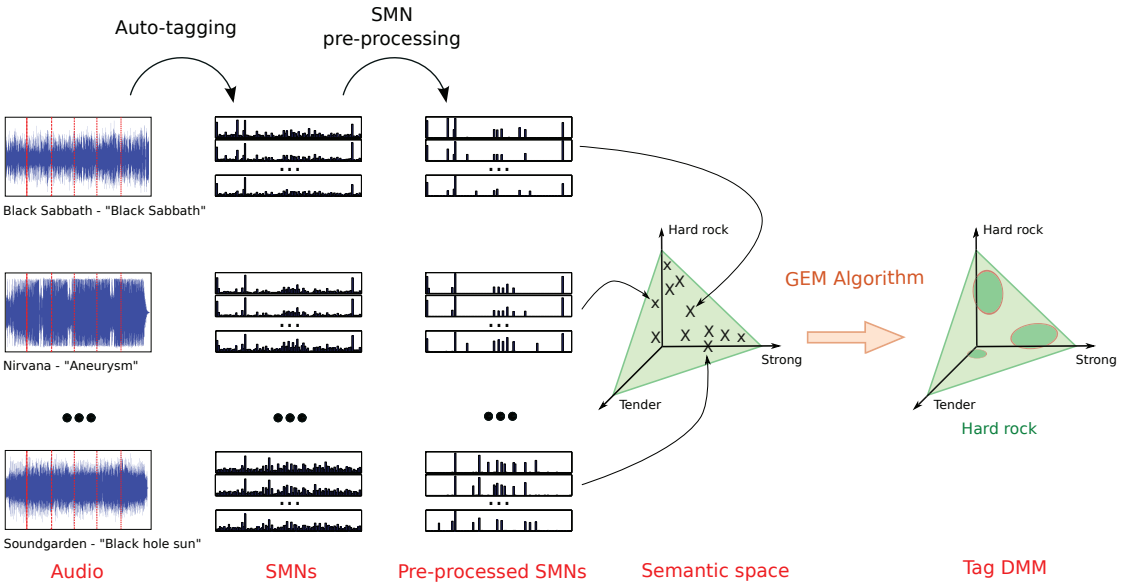


Figure 4.4. Learning a DMM context model for the tag "Hard rock". First, 5-second segments, extracted from each "Hard rock" song in the training set (the segments are extracted every 3 seconds, which, for clarity, is not precisely represented in the figure) are automatically annotated with semantic multinomials. Next, these SMNs are pre-processed, to highlight their peaks, as explained in the text. Lastly, the distribution of the SMNs in the semantic space is modeled by a Dirichlet mixture model using a generalized EM algorithm.

for the parameters that maximize the likelihood, which is intractable, this M-step generates a parameter estimate for which the likelihood is higher than the one in the previous iteration. This is known to be sufficient to guarantee convergence of the overall EM procedure [31]. To implement this M-step, we apply the Newton-Raphson algorithm, as in the work of Minka [64] who applied it for single component Dirichlet distribution (see Section 3.2) for more details). Unlike some other numerical algorithms the latter is more efficient since it does not require storing or inverting the Hessian matrix (the update for $\boldsymbol{\alpha}$ in Equation (4.8) can be computed more efficiently). Details are provided in Algorithm 1 (wherein $\mathbf{1}$ denotes the vector of all ones). Figure 4.4 illustrates the complete training procedure.

4.4.2 Contextual Auto-tagging

Once the contextual tag models, $P(\boldsymbol{\pi}|w_i;\Omega^{w_i})$, have been estimated for all $w_i, i = 1, ..., |\mathcal{V}|$ in the vocabulary $\mathcal{V}$, they can be used to annotate a new song.

First, for an unseen test song $\mathcal{X}$, (overlapping) 3-second segments are extracted from its audio signal, every 2 seconds. Since we are extracting segments from one song only at this stage (as opposed to extracting them from many during training), we extract slightly shorter segments than from training songs to generate a reasonably large and rich set of SMNs for the test song. Each segment is annotated with an SMN by the first-stage auto-tagger. This represents the test song as a collection of S semantic multinomials, i.e. $\mathcal{X}^{\boldsymbol{\pi}} = \{\boldsymbol{\pi}^1, ..., \boldsymbol{\pi}^S\}$,

where $\boldsymbol{\pi}^s = (\pi_1^s, ..., \pi_{|V|}^s)$ and S depend on the length of the song.

Given the set of SMNs representing $\mathcal{X}$, the most relevant tags are the ones with highest posterior probability, computed using Bayes' rule:

$$\theta_i = P(w_i|\mathcal{X}^{\boldsymbol{\pi}}) = \frac{P(\mathcal{X}^{\boldsymbol{\pi}}|w_i)P(w_i)}{P(\mathcal{X}^{\boldsymbol{\pi}})}, \tag{4.11}$$

where $P(w_i)$ is the prior of the i^{th} tag and $P(\mathcal{X}^{\boldsymbol{\pi}})$ the song prior. We assume a uniform prior, i.e. $P(w_i) = 1/|\mathcal{V}|$ for $i = 1, \ldots, |\mathcal{V}|$, to promote annotation using a diverse set of tags. The song prior, $P(\mathcal{X}^{\boldsymbol{\pi}})$, is computed as $P(\mathcal{X}^{\boldsymbol{\pi}}) = \sum_{j=1}^{|\mathcal{V}|} P(\mathcal{X}^{\boldsymbol{\pi}}|w_j)P(w_j)$. As in the work of Turnbull et al. [101], we estimate the likelihood term of Equation (4.11), $P(\mathcal{X}^{\boldsymbol{\pi}}|w_i)$ by assuming that song segments are conditionally independent (given w_i) and compensating for the inaccuracy of this naïve Bayes assumption by computing the geometric average of the S segment likelihoods:

$$P(\mathcal{X}^{\boldsymbol{\pi}}|w_i) = \left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_i)\right)^{\frac{1}{S}}. \tag{4.12}$$

Finally, collecting all posterior probabilities

$$\theta_i = P(w_i|\mathcal{X}^{\boldsymbol{\pi}}) = \frac{\left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_i)\right)^{\frac{1}{S}}}{\sum_{j=1}^{|\mathcal{V}|}\left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_j)\right)^{\frac{1}{S}}}, \tag{4.13}$$

provides the *contextual* multinomial $\boldsymbol{\theta}$, which can be used for annotation and retrieval tasks following the approach outlined in Section 4.2.

In Section 6.3, we will demonstrate that this contextual representation of songs improves annotation and retrieval for all the first-stage auto-taggers reviewed in Section 4.3.

CHAPTER 5

Combining Semantic and Content Information for Music Retrieval

Tags play a key role in the design of fully functional music retrieval systems. However, the huge scale reached by present-day music corpora has made the manual association of tags to songs infeasible. For this reason, several automatic tagging approaches have been recently proposed; the content-based auto-tagger paradigm described in the previous chapter is a first example. In alternative, competitive methodologies not related to the music content are the propagation of tags from similar songs, the use of users social behavior, or the use of Web mining techniques for searching tags into Web pages associated to the songs.

Nevertheless, state-of-the-art systems that automatically collect tags can lead to noisy representations, and this may negatively affect the effectiveness of retrieval algorithms. For instance, automatic tagging systems need highly reliable training data (i.e. verified tag-song associations) in order to achieve good predictive models. Additionally, these models have to handle robustness issues such as parameter over-fitting or term normalization, which are typically due to data sparseness. The use of social behavior to obtain tags may also result in poor descriptors. Social tags are sometimes referred to as the "wisdom of the crowd" since they are assigned to the songs by a large number of non expert humans. However, if a vocabulary of tags is not defined in advance, a human user of these systems can annotate songs without restrictions. Therefore, tags can contain typos (e.g. "classikjazz", "mellou"), be redundant (e.g. "hip-hop"-"hiphop", "hard rock"-"hardrock"-"harder rock"), or be simply not useful for retrieval purposes (e.g. "favorite artist", "mistagged").

An alternative retrieval mechanism can be carried out directly on the acoustic content of the songs, without relying on a semantic representation. This was the initial and typical approach in MIR, where audio and music processing is applied to compute acoustic similarity relationships between songs of a collection. Such similarity connections can then be used to rank songs in a content-based music retrieval system [19]. However, in music applications

the problem of selecting an optimal similarity measure is difficult because of the intrinsic subjectivity of the task: users may not consistently agree upon whether or to what degree a pair of songs or artists are acoustically similar. In this case, tags can be used to leverage acoustic similarity because they contextualize a song – e.g. describing a historical period or a geographical area related to the song. Hence, both approaches have some limitations when taken singularly; a combined use of them may instead lead to a reduction of such disadvantages.

This chapter presents a general model for searching songs in a music collection where both content-based acoustic similarity and tags are combined in a statistical framework. That is, the goal is to partially overcome the limitations of these two descriptions by using them together in a unified graph-based representation. The approach is based on an application of a hidden Markov model (HMM) to represent a set of songs; documents relevant for a query are retrieved by a modified version of the Viterbi algorithm. Retrieval can be performed efficiently and does not require any pre-processing steps or parameter estimation (i.e. training step). The model is applicable in different music discovery scenarios, such as item-based recommendations, playlist generation, and qualitative re-ranking of fast retrieval approaches.

The remainder of this chapter is organized as follows. After a discussion of related work in Section 5.1, the model and the retrieval framework are defined in Sections 5.2 and 5.3. Lastly, Section 5.4 introduces the music representations used to run the experiments reported in Section 6.4.

5.1 Related Work

A considerable amount of research has been devoted to semantic music search and discovery engines. These systems allow queries consisting of natural languages, such as tags or song metadata (i.e. title, artist, year, etc.) and return songs that are semantically related to this query [20, 51] (see Section 2.2.8). Most of the state-of-the-art content-based approaches rely on the auto-taggers described in the previous chapter, since they provide an easy framework for tag-based search and discovery (see Section 4.2.3). Barrington et al. [6] compare a content-based approach with Apple iTunes Genius, which recommend music by mainly exploiting users behavior (i.e. collaborative filtering). They showed that, while Genius globally performs better, the content-based research system achieves highly competitive results in the "long tail" of undiscovered and little recommended songs.

The HMM-based framework discussed in this chapter aims at improving the state-of-the-art discovery technologies by combining tags with content-based acoustic similarity. Berenzweig et all. [9] show that it is important to combine both subjective and acoustic representation when dealing with music similarity. As regards the semantic descriptions, besides the content-based auto-taggers reviewed in Section 4.1, different alternative approaches have been proposed to collect tags; in particular, social tagging [54], Web mining [51] or tag propagation from similar songs [91], each with advantages and disadvantages [98]. As regards

acoustic similarity, an introduction to the state-of-the-art methodologies has been already reported in Section 2.2.2.

In the literature, approaches that merge different heterogeneous descriptions of music information were proposed by Slaney [90] for music classification, by Turnbull et al. [99] and Tomasik et al. [94] for semantic retrieval, by McFee and Lanckriet [63] for artist similarity, and by Wang et al. [106] for artist style clustering. These methodologies generally learn the parameters of a function that is used to join the different sources of information in order to provide the user with more subjective song descriptions and similarity relationships. Our approach is consistently different because it is built on a graph-based representation that combines different music descriptions, and it does not rely on additional processing or training to carry out retrieval. This point is particularly important in scenarios where it is difficult to collect reliable ground truth data for the training step. Moreover, rather than providing a more meaningful description of songs or better similarity relationships, our goal is to deliver a unified retrieval mechanism located on a higher level than acoustic and semantic representations considered individually. This means that it can also exploit these more sophisticated descriptions. A graph-based representation for music recommendation is also used by Bu et al. [14]; however, instead of a graph, they use a unified hypergraph to model and combine social media information and music acoustic-based content. This approach is an application of ranking on graph data [2] and requires learning a ranking function; again, it can be highlighted that we do not need training operations to perform retrieval. Additionally, [14] covers a wider application than semantic retrieval, since they focus on the design of a hybrid recommender which also considers user-profile data. However, we do not investigate the extension of our model in a hybrid recommendation scenario where user-profile is also taken into account.

Knees et al. [50] examine the effect of incorporating audio-based similarity into a tag-based ranking process, either by directly modifying the retrieval process or by performing post-hoc audio-based re-ranking of the search results. The general idea is to include in the ranking list songs that sound similar to those already collected through tag-only retrieval; hence, acoustic similarity is used as a correction factor that improves the ranking list. Again, our approach is different because tags and acoustic similarity are used together at the same time in the retrieval process (i.e. one representation is not used to correct the ranking scheme of the other).

Music descriptions are combined by exploiting the proprieties of hidden Markov models. As already introduced in Section 3.3, HMMs have been extensively used in many applications where the concept of *time* plays a key role, such as speech recognition as well as music processing and retrieval. To the best of our knowledge, this is the first application of HMMs in the task of multimodal retrieval, where music content and tags are combined into a single framework.

The HMM-based framework is proposed as the core component of a semantic music discovery engine. However, the approach is very general and can be used in other scenarios as

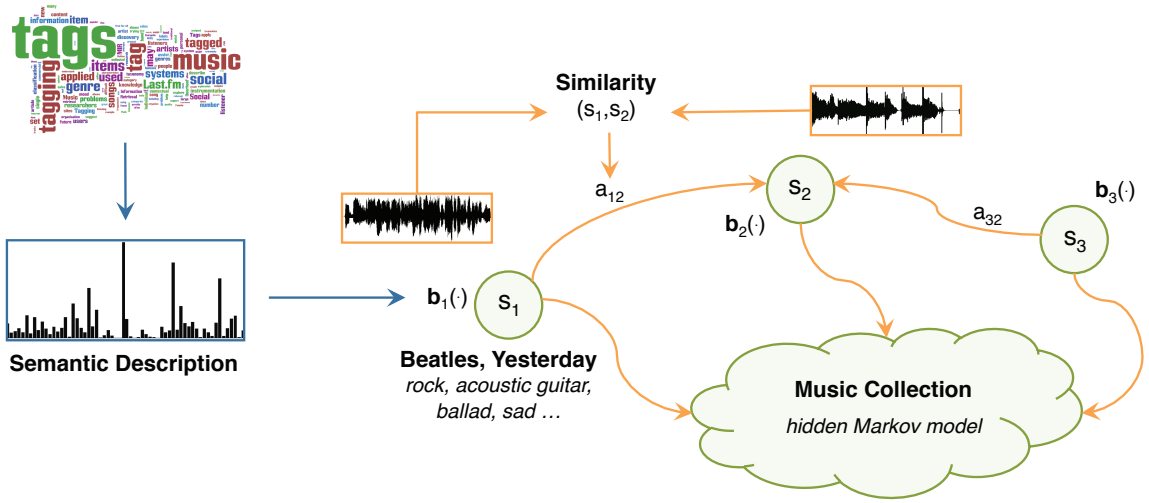


Figure 5.1. General structure of the model: each song included in a music set is represented by a state and is described by a set of tags. States are linked together by edges weighted according to acoustic similarity between the songs.

well as with other media, such as images and videos. The main requirement of the model is that a medium may be represented by semantic descriptors, and that it is possible to compute a similarity measure between digital items. In particular, image and video retrieval seem potential scenarios where the proposed framework may be applied (e.g. see [1, 82, 96]).

5.2 Statistical Modeling of a Music Collection

The goal of semantic music search and discovery engines is to retrieve a list of songs in response to the description of a user information need. In the music domain, such user information needs can be described either directly by a general semantic indication, such as the tag "rock", or indirectly by a seed song, such as the audio content and the set of tags assigned to "My Sharona". In both cases, it can be assumed that the goal of the user is to *observe* consistently the fulfilment of his information need during the time of his access to the music collection. In particular, a retrieval system should be able to return a list of music items which are acoustically coherent and pertinent to the request and, at the same time, are ranked according to their relevance to one or more semantic labels that give context to the user information need.

The methodology here discussed aims at providing a formal and general model to retrieve music documents by combining acoustic similarity and the semantic descriptions given by tags. That is, the goal is to propose a model that encompasses both content-based similarity and context-aware descriptors. To this end, HMMs are particularly suitable because they allow us to model two different sources of information. In fact, HMMs represent a doubly embedded stochastic process where, at each time step, the model performs a transition to a new state according to transition probabilities and it emits a new symbol according to

emission probabilities.

Thus HMMs can represent either content and context information, under the following assumptions [70]:

- if each state represents a song in the collection, acoustic content-based similarity can be modeled by transition probabilities;
- if the emitted symbols are semantic labels, the context that describes each state can be modeled by emission probabilities.

A suitably built HMM can be used to address the examples provided at the beginning of this section. On the one hand, the model can be used to create a path across songs while observing, for a defined number of time steps, the semantic label "rock". On the other hand, the model can start the path from the state associated with the song "My Sharona" and proceed to new states while observing the semantic labels associated to the seed song. In both cases, the songs in the path are likely to have a similar content because of transition probabilities and are likely to be in the same context because of emission probabilities.

Since states of a HMM are not directly observable, the paths across the songs need to be computed using a decoding step, which highlights the most probable state sequence related to a sequence of observations. A graphical representation of the model is depicted in Figure 5.1; songs are linked together through edges, which are weighted according to the similarity between the audio signals. In addition, each state is also described by a set of tags, which contextualize each song.

5.2.1 Definition of the HMM

As introduced in Section 3.3, a HMM modeling a collection of tagged songs can be formally defined through a set of parameters. In particular, a HMM λ is defined by:

- the number of songs N in the collection, each song represented by a state of the HMM. Individual states are denoted as $\mathcal{S} = \{S_1, ..., S_N\}$, and the state at time t as $s(t)$;
- the number M of distinct tags that can be used to describe a song. The set of symbols is defined by a finite size vocabulary $\mathcal{V} = \{w_1, ..., w_M\}$;
- the state transition probability distribution $\boldsymbol{A} = \{a_{ij} \mid i, j = 1, ..., N\}$, which defines the probability of moving from state i to state j in a single step. This distribution is related to the direct acoustic similarity between songs; therefore a_{ij} depends on the audio similarity between S_i and S_j;
- the emission symbol probability distribution $\boldsymbol{B} = \{b_j^1, ..., b_j^M \mid j = 1, ..., N\}$, which represents the probability that a tag $w \in \mathcal{V}$ is associated with song j. Probability values represent the strength of the relationships *song-tag*, which is sometime indicated as *affinity* value. The vector $\boldsymbol{b}_j(\cdot)$ represents the semantic description of the general song j;

- the initial state distribution $\boldsymbol{\pi} = \{\pi_1, ..., \pi_N\}$, which represents the probability of starting a path from state S_i, with $i = 1, ..., N$. Unlike the standard definition of HMMs, the initial state distribution is computed dynamically at retrieval time, since it is strictly connected to the type of information need, as described in Section 5.3.1.

We use the compact notation $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ to denote the HMM; the parameters N and M are inferred from the probability distributions.

Common approaches for computing acoustic similarity usually give a positive value for each pair of songs, implying $a_{ij} > 0 \ , \forall \ i, j$. However, with the aim of improving scalability, at retrieval time we consider each state to be connected directly to only the R acoustically most similar songs in the collection, while the transition probabilities with all the other states are set to zero. Section 6.4.2.1-(a) provides a more accurate analysis about the way the value of R affects the retrieval performances. Self-transitions are set to zero as well, because self-similarity is not a relevant factor in retrieval applications. Both transition and emission probabilities are normalized to satisfy the stochastic propriety of HMMs, that is

$$\sum_{i=1}^{N} a_{ij} = 1 \ , \quad \text{and} \quad \sum_{k=1}^{M} b_j^k = 1 \ , \quad \text{with} \quad j = 1, ..., N \ . \tag{5.1}$$

Because of these steps, transition probabilities are usually not symmetric (i.e. $a_{ij} \neq a_{ji}$), which, at least in the case of music, is a reasonable assumption. In fact, a garage band can play a cover of a Beatles song which could be considered very similar to the original (because it is a cover); however, the version recorded by the Beatles might not be similar to any of the unknown band.

After setting all the parameters, the HMM can be used to generate observation sequences, where the observed symbols are tags. Dually, well known algorithms can be used to decode the most likely state sequence of the model which has generated a given observation sequence.

5.3 Ranking the Songs

At retrieval time, the goal is to highlight a sub-set of songs that are relevant to a particular query, either expressed by tags or by a seed song. In the context of HMMs, the task can be related to Problem 2 of Section 3.3.1, that is:

"*given the observation sequence* $\boldsymbol{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_T\}$*, and the model* λ*, find a corresponding state sequence* $\{s(1)^*, ..., s(T)^*\}$ *which is optimal in some sense*".

The description of the user information need is represented by the observation sequence $\boldsymbol{O}$; the value of T defines the number of songs which are retrieved.

In HMM applications, this decoding problem can be solved using the Viterbi algorithm (see Section 3.3.2). The latter acts as a max-sum algorithm and searches efficiently in the space of possible paths to find the optimal one (i.e. the most probable one). The cost of the process grows only linearly with the length of the desired ranking list; this means that retrieval

can be performed efficiently simply by keeping the value of T relatively small (i.e. shorter ranking lists). The algorithm is composed of a forward computation to find the maximization for the most probable path, and by a backward computation to decode the sequence of state. Although the general structure of the original algorithm reported in Section 3.3.2 has been maintained, we introduce some modifications; the details are provided in Algorithm 2. The overall time complexity is $\mathcal{O}(TN^2)$, whereas the space complexity is $\mathcal{O}(TN)$.

Algorithm 2 defines $OBS_j(\cdot)$, a general function related to the type of query (i.e. tags or seed-song) which specifies how the semantic description is considered during the retrieval process. This function plays the role of observations in typical HMM applications. Motivations and details are discussed in Section 5.3.1.

The recursion step expressed in Equation 5.6 introduces a variation of the role of transition probabilities. In fact, because of the structure of the model, it could happen that the optimal path enters a loop between the same subset of songs or, in the worst case, jumps back and forth between two states. Clearly, this is a problem because the retrieved list would present the same set of songs multiple times. Moreover, the loop could be infinite, meaning that the algorithm cannot exit from it and the retrieval list would only be composed by very few items. We addressed this problem by considering a penalty factor that is applied to the transition probabilities when they have already been chosen during the forward step. In fact, when a transition is chosen, we decrease the corresponding probability value by factor η (we used $\eta = 10$), as shown in the third line of Equation 5.6; this makes it unlikely for the states sequence to pass again through that transition. Attenuation is carried out temporarily, meaning that it affects the structure of the model only during the current retrieval operation. It should be noted that after this modification there is no guarantee that the path is globally optimal; however, what is relevant to our aims is that the path has a high probability of being relevant to the query from both content and context information and, at the same time, covers a large part of the collection.

Another issue that has to be addressed is a limitation in the structure of standard HMMs. Because of the first-order Markov chain assumption, HMMs are generally poor at capturing long-range correlations between the observed variables, that is between states separated by many steps in the decoding sequence [12]. Earlier experiments showed that this limitation involved a decrease in precision when decoding long paths. In order to tackle this problem, we consider the retrieval process composed by many shorter sub-steps, each one retrieving a sub-sequence of the final ranking list. When one of the sub-sequences is delivered, the following retrieval sub-step restarts from the last song previously suggested and is performed only on the songs not yet included in the ranking. Given the locality of the approach, in this way we attempt to keep constant the correlation between the query and the retrieved songs along the whole list. Additionally, splitting the retrieval in sub-steps allows the system to better interact with the final users, who can obtain the ranking list of songs incrementally, instead of having to wait for the whole computation. For instance, the model can be set to deliver enough songs at each sub-step to fill a Web page; in this way the remaining items

Algorithm 2 The Viterbi-Like Algorithm for HMM-based Retrieval

1: **Input**: the model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, the vocabulary of tags $\mathcal{V}$ and the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_T\}$, such that for each $t = 1, ..., T$,

$$\boldsymbol{o}_t(\cdot) = \begin{cases} \{(o_t^1, ..., o_t^k) \mid o_t^y = w \in \mathcal{V} \ \text{ for } \ y = 1, ..., k\} & \text{if } k\text{-tag query-by-description,} \\ \boldsymbol{b}_q(\cdot) \quad \text{if query-by-example with song } q. \end{cases} \tag{5.2}$$

2: Define the KL divergence $D_{KL}(\boldsymbol{x} \parallel \boldsymbol{y})$ between the general discrete probability distributions $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$D_{KL}(\boldsymbol{x} \parallel \boldsymbol{y}) = \sum_{i=1}^{|x|} x_i \cdot \log \tfrac{x_i}{y_i} \ , \quad \text{where} \quad |\boldsymbol{x}| = |\boldsymbol{y}| \ . \tag{5.3}$$

3: Define $OBS_j(t)$ with $j = 1, ..., N$ and $t = 1, ..., T$, according to $\boldsymbol{O}$:

$$OBS_j(t) = \begin{cases} b_j(o_t^1) \ \cdot \ b_j(o_t^2) \cdot \ ... \ \cdot b_j(o_t^k) & \text{if query-by-description,} \\ \frac{1}{D_{KL}(\boldsymbol{b}_j(\cdot) \| \boldsymbol{b}_q(\cdot))} & \text{if query-by-example.} \end{cases} \tag{5.4}$$

4: **Initialization**: for $j = 1, ..., N$

$$\delta_1(j) = \pi_j \cdot OBS_j(1) \ , \qquad \psi_1(j) = 0 \ . \tag{5.5}$$

5: **Recursion**: for $t = 2, ..., T \ , \ \ j = 1, ..., N$

$$\begin{aligned} \delta_t(j) &= \max\nolimits_{1 \le i \le N} [\delta_{t-1}(i) \ \cdot \ a_{ij}] \ \cdot \ OBS_j(t) \ , \\ \psi_t(j) &= \arg\max\nolimits_{1 \le i \le N} [\delta_{t-1}(i) \ \cdot \ a_{ij}] \ , \\ a_{uj} &= \tfrac{a_{uj}}{\eta} \quad \text{with} \quad u = \psi_t(j) \ , \ \ \eta = 10 \ . \end{aligned} \tag{5.6}$$

6: **Termination**:

$$s(T)^* = \arg\max\nolimits_{1 \le i \le N} [\delta_T(i)] \ . \tag{5.7}$$

7: **Decoding**: for $t = T - 1, T - 2, ..., 1$

$$s(t)^* = \psi_{t+1}(s(t+1)^*) \ . \tag{5.8}$$

8: **Output**: the ranking list $\{s(1)^*, ..., s(T)^*\}$.

may then be retrieved while the user is already checking the first delivered results.

5.3.1 Querying the Model

In the interaction with music discovery and search engines, among the different paradigms of Section 2.2.8, we consider two different types of query: tags (*query-by-description*) and seed

song (*query-by-example*). The proposed framework can easily handle both cases; however, according to the topology of the query, some parameters need to be set differently.

In the *query-by-description* scenario, the model has to rank the songs according to their relevance with the provided tags. The query may be composed of a single tag (e.g. "rock") or a combination of tags (e.g. "mellow", "acoustic guitar", "female lead vocals"). In this case, the observation sequence at iteration t, with $t = 1, ..., T$, is the vector $\boldsymbol{o}_t = \{o_t^1, \cdots, o_t^k\}$, where k is the number of tags in the query. We consider the tags in the query having the same importance; however, tags may be weighted in relation to the query string. For example, it is possible to give more weight to the words that appear earlier in the query string as is commonly done by Internet search engines for retrieving Web documents. However, we do not explore such extension in this work. We decide to set the initial state probability equal for all the songs in order to let the algorithm decide the beginning of the retrieved list. Therefore, the function $OBS_j(t)$ of Algorithm 2 is defined as

$$OBS_j(t) = b_j(o_t^1) \ \cdot \ b_j(o_t^2) \cdot \ ... \ \cdot b_j(o_t^k) \ , \tag{5.9}$$

for a generic state j. It can be noted that observations may be composed of the same tags for all the time steps, or they may change over time (e.g. starting with "rock" for the first iterations, going on with "danceable" in the following instants, and so on).

In the *query-by-example* scenario, the system is required to provide the user with a list of songs similar to a seed song q. The initial state distribution is forced to be 1 for the state representing the seed song and 0 for all the others[1]. The observation sequence to be decoded is modeled as the tags of the seed song, that is $\boldsymbol{b}_q(\cdot)$; so, in this case $OBS_j(t)$ is defined as the inverse of the Kullback Leibler (KL) divergence between the semantic description of a chosen state and $\boldsymbol{b}_q(\cdot)$ [52]. The choice of the KL divergence aims at generalizing the terms used for the tags, because it is related to the similarity of concepts associated with the tags rather than the pure distance between lists of tags. We consider the inverse because the goal is to maximize the probability when the divergence is small. Therefore,

$$OBS_j(t) = \frac{1}{D_{KL}(\boldsymbol{b}_j(\cdot) \parallel \boldsymbol{b}_q(\cdot))} \ , \quad \text{where} \quad D_{KL}(\boldsymbol{b}_j(\cdot) \parallel \boldsymbol{b}_q(\cdot)) = \sum_{i=1}^{M} b_j(i) \cdot \log \frac{b_j(i)}{b_q(i)} \ , \tag{5.10}$$

for the generic state j and the seed q[2]. In this case, observations do not change over time t, since they are only linked to the tags of the seed song.

5.3.2 Improving Ranking with the Negative Requirements

The ranking list delivered by the model is a list of songs positively associated with the description of the information need. However, the quality of the retrieval can be improved by

[1] We assume that the seed song q is stored in the collection. Possible extensions to scenarios where q does not belong to the corpus are not treated in this work.

[2] In preliminary experiments we also tried to compute the similarity between two songs as the negative exponentiation of their KL divergence; however, this solution led to worse empirical results.

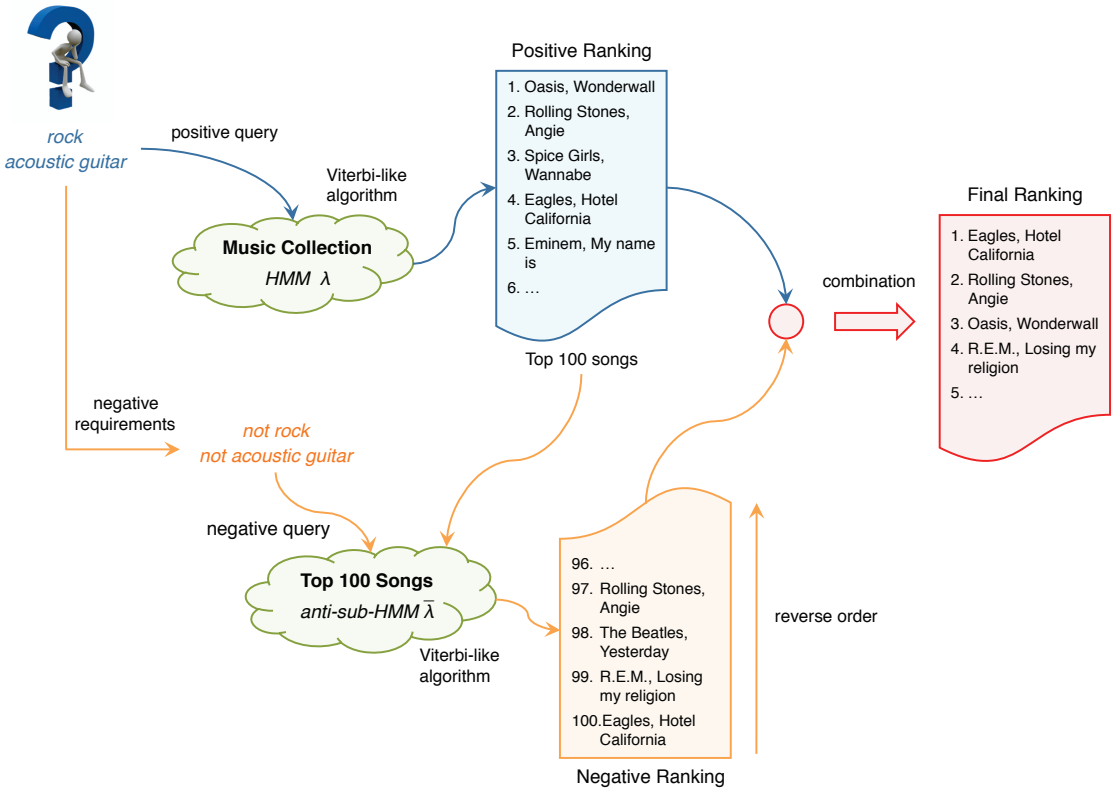


Figure 5.2. HMM-based retrieval: the user submits a query (tags or seed-song) to the system which ranks the song positively associated with the query. Later, the first positions of the list are re-ranked according to the negative information carried by the query. At the end, the system provides the user with a final ranking list obtained by combining both positive and negative results.

also considering the negative information, which is latent in the query. In fact, when a user is searching for "rock", his main expectation is to find songs at the top of the ranking lists associated with this tag, both semantically and acoustically. Additionally, he also expects the strongly "not rock" songs to be lower. Then, to increase the satisfaction of the user, the "not-rock" items should be discarded from the top of the list. Similarly, in the query-by-example scenario, if a user is searching for "My Sharona", all the songs not semantically related to it should be ranked at the bottom.

The HMM-based framework can be easily extended to exploit the negative information expressed by the query; the idea is to sort the songs also according to their relevance to the anti-tags, which are semantically defined as the NOT version of a tag (i.e. "not rock", "not tender", "not acoustic guitar", etc.). The set of anti-tags for a song represents its anti-semantic description. However, the antonyms (or at least the NOT versions of the tags) are not always explicitly provided for all the tags in a vocabulary. For this reason, the negative description for each song j is represented as

$$\bar{\boldsymbol{b}}_j(\cdot) = 1 - \boldsymbol{b}_j(\cdot) \ , \quad \text{with} \quad j = 1, ..., N \ . \tag{5.11}$$

The extended approach refines the ranking output by the positive model, through a

combination with the anti-tags retrieval. In particular, we consider the first 100 songs of the positive ranking list, and we build an *anti-sub-HMM* $\bar{\lambda}$ using only the data of these songs. The emission probability distribution of each state j in $\bar{\lambda}$ is the anti-semantic description $\bar{\boldsymbol{b}}_j(\cdot)$ defined in Equation 5.11. The choice of using the first 100 songs depends on the fact that we want to use the negative retrieval only to refine the ranking delivered to the user. Moreover, this also allows for minimizing the additional computational time. With this configuration, we run Algorithm 2 with the negative query in order to achieve the negative ranking list. Ideally, in this list the songs relevant for the anti-tags are located at the top of this list. Conversely, the bottom should be the place where the songs positively associated with the original query are found. For this reason, we consider this negative ranking in the reverse order.

Lastly, the two ranking lists (positive and negative reversed) are combined to obtain the final list. The combination is performed using the ranking values. In particular, we define a score by computing the mean of the two ranks (e.g. a first and a fifth rank gave a score equal to 3); the songs are then sorted in ascending order according to this value to achieve the final ranking list delivered to the user. Figure 5.2 shows the complete retrieval process, from the query to the two retrieval steps (positive and negative), till the final result.

5.4 Music Representation

The HMM-based engine provides a general framework for combining different representations of the music items, in particular acoustic similarity and tags. This section overviews the methodologies used to describe the songs in the experiments reported in Sections 6.4.1 and 6.4.2; more details can be found in the corresponding references. It is important to note that the proposed framework can be used with a number of alternative music content descriptions discussed in the previous sections. The approaches here described represent only the experimental setup we built to validate the proposed model.

5.4.1 Audio Features

To define similarity relationships between songs we mostly use the timbre descriptors introduced in Chapter 3. In particular, when having access to the audio clips of a dataset we use MFCCs (see Section 3.1.1), while when having access only to a list of song titles (or song IDs) we use the ENT features alone (see Section 3.1.2).

In particular, MFCCs are computed on half-overlapping windows of 23 ms using 13 components. We extract MFCCs for at most 4 minutes of music taken from the middle of the signal; at the end, for each song we randomly sub-sample $15,000$ feature vectors in order to obtain a same-size representation of each item in the collection. This last operation reduces the computational cost without affecting the precision of the similarity computation [101]. ENT features follows the standard Echo Nest API definitions described in Section 3.1.2. Additionally, we append to each feature vector of both representations the corresponding first

and second instantaneous derivatives; this leads to the final 39-dimensional delta-MFCCs and 36-dimensional delta-ENTs.

In the case of having access to the audio clips we also include Fluctuation Patterns (FP) to describe the amplitude modulation of the loudness per-frequency band [76]. FPs describe characteristics of the audio signal which are not captured by the spectral representation, such as the periodicity of the music signal. Following the implementation outlined by Pampalk [74], a FP is computed by: (i) cutting the spectrogram into segments, (ii) using an FFT to compute amplitude modulation frequencies of loudness (range 0-10Hz) for each segment and frequency band, (iii) weighting the modulation frequencies based on a model of perceived fluctuation strength, (iv) applying filters to emphasize certain patterns and smooth the result. The resulting FP is a 12 (frequency bands according to 12 critical bands of the Bark scale) times 30 (modulation frequencies, ranging from 0 to 10Hz) matrix for each song. We compute FPs on the whole music length over half-overlapping segments of 3 seconds using a 20-band spectrogram. The combined use of MFCCs and FPs is very common in MIR, especially for hubness reduction [37].

5.4.2 Acoustic Similarity

Acoustic similarity is computed as a distance measure between the content features of a song and all the other items in the music corpus. A number of approaches have been proposed for this task (see Section 2.2.2); in this work we use two distances computed over different audio features and, when possible, we derive a single similarity measure through a weighted combination between them[3]. Both approaches are fast in computation and easy to implement. Since we aim at showing the general performances of the model, and not a particular configuration, we prefer such aspects to the greater precision achieved by other competitive models requiring a more sophisticated parameter estimation, and whose implementation is more error-prone.

Timbral similarity is computed by representing the timbre descriptors of each song (i.e. delta-MFCCs or delta-ENTs) as the single Gaussian (full covariance) with the maximum likelihood of fitting the song's features. The distance between two Gaussians is a symmetric version of the KL divergence (i.e. $D_{KL}(x \parallel y) = D_{KL}(y \parallel x)$, rescaled to turn into a similarity value (i.e. negative exponentiation of each divergence). Since the standard KL divergence does not lead to symmetric values, the symmetry is achieved by summing the two divergences, i.e. $D_{KL}(x \parallel y) + D_{KL}(y \parallel x)$ [60].

In contrast, similarity of FPs is computed as the re-scaled Euclidean distance between two FPs [74].

To obtain an overall similarity measure $sim(x, y)$ between the two songs x and y, all the

[3]We implemented by ourselves the described algorithms for acoustic similarity by using the details of the papers they refer to.

described similarity values are combined by a simple arithmetic weighting:

$$sim(x, y) = 0.7 \cdot z_G(x, y) + 0.3 \cdot z_{FP}(x, y) \ , \tag{5.12}$$

where $z_G(x, y)$ and $z_{FP}(x, y)$ are the value of the timbre-based and FP-based similarity after z-normalization. We set the weighting coefficients as in [50].

Lastly, we used the combination shown in Equation 6.4 only when we had access to the audio clips of a dataset. Conversely, in cases where we had the ENT features only, the similarity computation refers to the single Gaussian-based timbral similarity alone.

5.4.3 Tags

We use two different automatic strategies to collect tags: content-based auto-taggers and social tags. We separately address these two different representations; however, they could be combined into a single richer one as described by Turnbull et al [99]. For both approaches, we consider a vocabulary $\mathcal{V}$ of $|\mathcal{V}|$ distinct tags w_i, for $i = 1, ..., |\mathcal{V}|$. At the end, each song is described by a vector of size $|\mathcal{V}|$ of tags relevance which measure the strength of the relationships tag-song. This vector (normalized to one) represents the state observations in the HMM-based retrieval framework (note that $M = |\mathcal{V}|$).

5.4.3.1 Content-based Auto-Taggers

Auto-taggers have been extensively described in Chapter 4. Among all the approaches, we consider the combination GMM-DMM [66]: the Gaussian mixture model (see Section 4.3.1) is used to model the acoustic content, while the Dirichlet mixture model introduced in Section 4.4 refines the predictions considering the contextual relationships. We use this combination because it is built on two generative approaches, leading to better predictions in case of training with a weakly labeled corpus (as reported in the experiments described later in Section 6.3.2.3).

5.4.3.2 Social Tags

Social tags were gathered from Last.fm, as available on November 2010. The users of this service can assign tags to the songs they are listening to, so these tags provide a valuable source of information on how people perceive and describe music. However, the data collection process is noisy since it involves a large community of non-expert fans annotating music with an unconstrained vocabulary of free-text tags. Therefore, many tags may be redundant and inconsistent. Additionally, the vocabulary of tags that emerges from the community tends to be very simple and focused on social, rather than acoustic aspects of the music.

For each song of the corpus, we collected two lists of social tags using the Last.fm public data sharing AudioScrobbler website [4]. We gathered both the list of tags related to a song

[4] `http://ws.audioscrobbler.com/2.0/`

and the list of tags related to an artist. The overall list of scores is given by summing the scores in both lists.

The retrieval framework proposed works in the presence of a finite size tag vocabulary. Thus, the gathered social tags have to be mapped into the equivalent classes in our vocabulary. In order to match the largest part of the classes, we pre-process the social tags through stemming (e.g. joining "rock", "rockers", "rocking"), noise reduction (i.e. joining "r&b", "rhythm and blues", "r & b"), and synonyms detection (e.g. annotating with "down tempo" if the social tag is "slow beat"). When annotating a song, a tag in the vocabulary that correctly matches a social tag takes the corresponding Last.fm-based score; otherwise, it takes a zero value. Therefore, if no social tags of a song matches any tags in the vocabulary, that song is represented by a uniform description where all the tags share a same relevance.

CHAPTER 6

Experimental Evaluation

One of the challenges of designing a music search engine is how to evaluate the novel methodology. Although several efforts have been made within the MIREX campaigns [32], data of past campaigns are not always freely available for testing new approaches, because of copyright issues. Ideally, the results of a music retrieval system should be evaluated by humans, in order to consider the subjectivity of the music similarity concept; however, human evaluation is a time consuming task which cannot performed for each experiment.

Over the years, researchers of the MIR community have provided some standard datasets which can be used as evaluation ground truth. Besides providing the music files or some audio descriptors, the songs of these collections are annotated with tags taken from a finite size vocabulary. Annotations are collected through a sort of interaction with humans, leading to reliable annotations for evaluation purposes. In fact, these annotations summarize human opinions which also take into account the subjective nature of the music perception.

This chapter reports the experimental evaluation that we performed to validate the models proposed in Chapters 4 and 5. In particular, we used the two standard datasets presented in Section 6.1 as experimental framework; in addition, the metrics used as a measure of effectiveness for the annotation and retrieval tasks are described in Section 6.2. Lastly, experiments and discussions for the DMM and the HMM-based retrieval frameworks are presented in Sections 6.3 and 6.4 respectively.

6.1 Music Datasets

We carried out all the experiments using two human-annotated music collections: CAL500 and Swat10k.

6.1.1 CAL500 Database

The Computer Audition Lab (CAL500) dataset was released by Turnbull et al. [100]; it consists of 502 popular songs of Western music by as many different artists. This set was chosen to maximize the acoustic variation of the music while still representing some familiar genres and popular artists. Through a controlled survey, each song has been tagged by at least 3 human annotators using a semantic vocabulary of 149 tags. The vocabulary is diverse and spans genres, instruments, vocal characteristics, acoustic characteristics, emotions and song usages. The CAL500 dataset provides binary annotations, which are 1 when a tag applies to the song (i.e. at least 2 subjects voted for the tag) and 0 otherwise. It is a strongly labeled data set in that both positive (1) and negative (0) associations have been verified.

The dataset is released as a corpus of audio clips (i.e. MP3 files); this allows for greater personalization when extracting the audio descriptors. Therefore, in all the experiments with CAL500, each model previously discussed is used with the audio features (or combinations of them) introduced in the corresponding section.

6.1.2 Swat10k Database

Swat10k [93] is a collection of 10,870 songs from 4,597 different artists, weakly labeled from a vocabulary composed of 137 "genre" tags and 416 "acoustic" tags. Each song is labeled with 2 to 25 tags. The song-tag associations for this data set have been mined from the Pandora website. As a result, Swat10k is a weakly labeled data set: while the presence of a tag means musicologists involved with Pandora's Music Genome Project assigned the tag, the absence of a tag does not imply whether it applies or not. In any case, since Pandora claims that their musicologists maintain a high level of agreement, the positive song-tag annotations are considered highly *objective*.

As copyright issues prevent us on obtaining all Swat10k audio clips, we represent their audio content using the delta-ENT features discussed in Section 3.1.2 alone.

6.2 Evaluation Metrics

A large part of the experiments performed pertains to retrieval operation and ranking list analysis. However, when evaluating auto-taggers the annotation task should also be taken into account.

6.2.1 Annotation

Annotation performance is measured following the procedure described by Turnbull et al. [101]. Test set songs are annotated with the most likely tags in their semantic or contextual multinomial (i.e. the number of most likely tags considered is set according to the size of the vocabulary). Annotation accuracy is measured by computing precision, recall and F-score for each tag, and then averaging over all tags. We compute annotation metrics on a *per-tag*

basis, as our goal is to build an automatic tagging algorithm with high stability over a wide range of semantic tags. *Per-song* metrics may get artificially inflated if a system consistently annotates songs with a small set of highly frequent tags, while ignoring less common tags. Per-tag precision is the probability that the model correctly uses the tag when annotating a song. Per-tag recall is the probability that the model annotates a song that should have been annotated with the tag. F-score is the harmonic mean of precision and recall. Precision, recall and F-score for a tag w are defined as:

$$p = \frac{|w_C|}{|w_A|} \ , \quad r = \frac{|w_C|}{|w_H|} \ , \quad f = \frac{2}{\frac{1}{p} + \frac{1}{r}} \ , \tag{6.1}$$

where $|w_H|$ is the number of songs that are annotated with w in the ground truth, $|w_A|$ is the number of songs automatically annotated by the model with the tag w, and $|w_C|$ is the number of times w is correctly used. In case a tag is never selected for annotation, the corresponding precision (that otherwise would be undefined) is set to the tag prior from the training set, which equals the performance of a random classifier.

6.2.2 Retrieval

A music search engine responds to a query by delivering a ranking list of documents sorted by their relevance to such query. Retrieval performances are measured by searching the relevant documents for a query along the ranking list, and averaging the measures over all the queries. We report some standard information retrieval metrics (see Chapter 8 of [62] for more details):

- **Precision at k (Pk)**, which measures how many good results there are at the beginning of the list, reporting the fraction of relevant documents in the top-k positions. The values of k used in the experiments are 1, 3, 5, 10 and 20;

- **Mean Reciprocal Rank (MRR)**, which averages the inverse of the rank of the first correct answer for each single query. In particular, for a query set $Q = \{q_1, ..., q_{|Q|}\}$, the mean reciprocal rank is defined as

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank } q_i} \ , \tag{6.2}$$

  where rank q_i denotes the rank of the first relevant document for the i^{th} query. Basically, MRR measures where the first relevant document is placed in the ranking list, and it is upper bounded by 1 (i.e. a relevant document ranked as first);

- **Mean Average Precision (MAP)**, which averages the precision at each point in the ranking list where a song is correctly retrieved. Precision is defined as the fraction of retrieved documents that are relevant. For each query, the average precision is the average of the precision value obtained for the set of top k documents existing after

each relevant document is retrieved. In particular, assuming that a general query $q_i \in Q = \{q_1, ..., q_{|Q|}\}$ has the set of relevant documents $\{d_1, ..., d_{m_i}\}$, the mean average precision overall Q is defined as

$$\text{MAP} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{m_i} \sum_{k=1}^{m_i} \text{Precision } (R_{ik}) \ , \tag{6.3}$$

where R_{ik} is the set of ranked retrieval results from the top result until you get to document d_k. When a relevant document is not retrieved at all, the precision value in the above equation is taken to be 0. Note that MAP is equal to MRR when a query has only one relevant document;

- **Area under the curve ROC (AROC)**, which is computed by integrating the ROC curve (ROC stands for "Receiver Operating Characteristics"). The ROC curve plots the true positive rate against the false positive rate of the ranking list. The area under this curve is upper bounded by 1; random guessing would result in an AROC of 0.5.

6.3 DMM Experiments

This section describes the evaluation carried out to test the effectiveness of the DMM framework presented in Chapter 4. In particular Section 6.3.1 introduces the experimental setup, while Section 6.3.2 reports all the results achieved for annotation and retrieval.

6.3.1 Experimental Setup

To evaluate the benefit of context modeling with DMMs, we first combine each of the semantic tag models described in Section 4.3 with our DMM-based context model, as depicted in Figure 4.3. For each combination, we compare the annotation and retrieval accuracy to the one obtained by the semantic tag models alone (i.e. without DMM).

Second, we compare our generative, DMM-based approach to context modeling with several recently proposed discriminative approaches based on SMNs. Most of these discriminative approaches were developed and evaluated in combination with a specific auto-tagger. In our experiments, we provide a more general evaluation. As for the evaluation of DMM, we combine each of these context models with all four semantic tag models and report the resulting annotation and retrieval performance. In particular, we compare with the following approaches:

- **Support Vector Machine (cSVM):** Ness et al. [72] propose to make tag predictions using SVMs based on semantic multinomials. In their work, the SMNs are computed by auto-taggers based on SVMs as well. We refer to this discriminative approach as cSVM (context-level SVMs). To train cSVM, we first re-scale the SMNs so that the minimum tag weight is 0 and the maximum 1 (min-max normalization). We implement cSVM using LibSVM [22];

- **Boosting (cBST):** Similar to cSVM, this discriminative approach adopts a set of binary classifiers to predict tags based on semantic multinomials. In the work of Bertin-Mahieux et al. [10], the semantic tag models generating the SMNs were also based on boosting, as described in Section 4.3.3. The classifiers are based on boosting and we refer to this model as cBST (context-level boosting);

- **Ordinal Regression (OR):** Yang et al. [110] also propose a discriminative approach to predict the relevance of a tag, based on semantic multinomials obtained with an ordinal regression model that is similar to their context model. They model several levels of relevance (as opposed to binary classification) by formulating tag prediction as an ordinal regression problem. For each tag w_i, training songs are assigned to one of 4 ordinal classes of decreasing relevance. The assignment is based on all tags training songs are annotated with and the empirically observed correlations of those tags with w_i (by computing Pearson coefficients over all training data). Moreover, before training a tag model, the training SMNs are pre-processed by removing the tags that are weakly correlated (Pearson coefficient 0.2 or lower) to the target tag and, thus, probably irrelevant for predicting it. Lastly, an ordinal regression model is trained for each tag w_i, using the listNet algorithm [109]. This results in $|\mathcal{V}|$ ordinal regression models. Given a new, unseen song, the relevance of each tag in the vocabulary $\mathcal{V}$ is computed based on these models. In our implementation, all the parameters have been set to the values reported in the work by Yang et al. [110].

Lastly, we also include in the comparison a lower bound (Rand) and an empirical upper bound (UpBnd) for annotation and retrieval performances. "Rand" samples words (without replacement) from a multinomial distribution parametrized by the word prior distribution $P(w_i)$, for $i = 1, ..., |\mathcal{V}|$. Intuitively, this prior stochastically generates annotations from a pool of the most frequently used words in the training set. In contrast, "UpBnd" uses the ground truth to annotate songs and shows the upper limit on what any system could achieve. However, since we require that each model uses a fixed number of words to annotate each song, if the ground truth annotation contains either too many or too few annotations, we randomly add words to or remove words from the songs semantic description.

6.3.2 Results

After discussing the pre-processing of SMNs for DMM in more detail, we provide annotation and retrieval results on the CAL500 and Swat10k data sets. Since we have access to all audio clips of the CAL500 corpus, we extract from the audio signal those features that the respective auto-tagging algorithms (GMM, CBA, SVM, and BST) were originally proposed with (see Section 4.3). Differently, in the experiments with the Swat10k dataset, all auto-tagging algorithms are trained and evaluated based on the Delta-ENT feature vectors only.

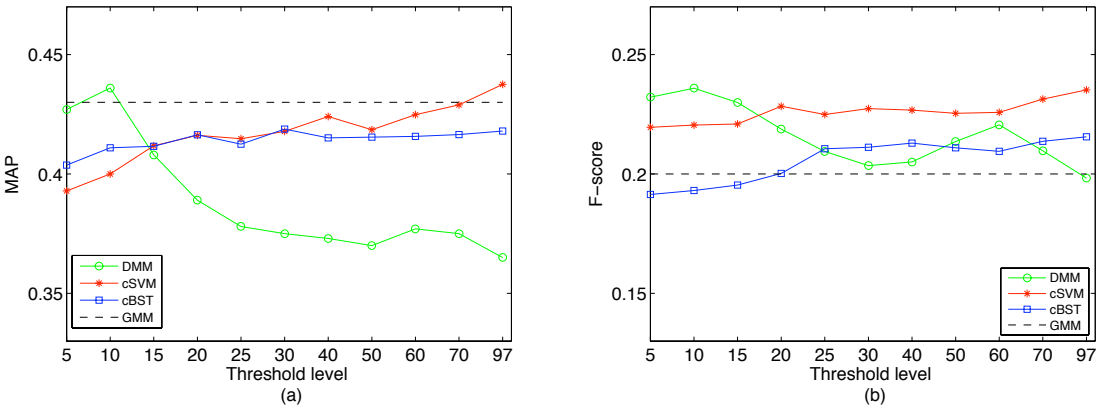


Figure 6.1. The effect of pre-processing SMNs before training DMM, cSVM and cBST models. The SMNs are generated by a GMM-based auto-tagger. (a) Retrieval performance (MAP) and (b) annotation performance (F-score). The dashed line represents the performance achieved without context modeling, i.e. using SMNs rather than CMNs.

6.3.2.1 Highlighting the Semantic Peaks for Training DMMs

The goal of the generative, DMM-based approach to context modeling is to estimate class-conditional densities that detect relevant contextual patterns in the SMNs while ignoring accidental tag co-occurrences as noise. Highlighting the peaks in the training SMNs is expected to aid this process by improving the detection of relevant co-occurrence patterns and reducing noise. As mentioned in Section 4.4.1, this is achieved by limiting each training SMN to its h most relevant peaks and reducing the weights of all other tags to very low values.

To evaluate the effectiveness of this pre-processing step, we first consider a simpler approach that selects the top-h peaks for all training SMNs (irrespective of their statistical properties). We vary the threshold h between 5 and $|\mathcal{V}|$ and analyze how this affects the annotation (F-score) and retrieval (MAP) performance on the CAL500 dataset, when learning a DMM context model over GMM-based SMNs. To accurately fit the DMM models, we restrict ourselves to the subset of 97 tags that have at least 30 training songs positively associated with them (11 genre, 14 instrument, 25 acoustic quality, 6 vocal characteristics, 35 emotion and 6 usage tags). We use 5-fold cross validation and annotate songs with the 10 most likely tags in the annotation task. For comparison, we also learn cSVM and cBST context models from the same pre-processed training SMNs.

The results, reported in Figure 6.1, indicate that DMM-based context modeling benefits from SMN pre-processing. The discriminative models (cBST, cSVM), on the other hand, perform worse when pre-processing training SMNs. Therefore, in the remainder of this section, SMN pre-processing is applied only for DMM training.

The kurtosis-based approach introduced in Section 4.4.1 adapts the pre-processing to the "shape" of the SMN. It selects more peaks for more uniform SMNs (lower kurtosis)

Semantic	Context	**Annotation**			**Retrieval**				
Level	Level	Precision	Recall	F-Score	P3	P5	P10	MAP	AROC
	Rand	0.240	0.102	0.140	0.239	0.241	0.243	0.276	0.504
	UpBnd	0.716	0.471	0.514	1.000	0.993	0.942	1.000	1.000
	alone	0.405	0.202	0.219	0.456	0.455	0.441	0.433	0.698
	cSVM	0.380	0.225	0.235	0.510	0.490	0.450	0.435	0.686
GMM	cBST	0.412	0.163	0.215	0.465	0.460	0.440	0.429	0.689
	OR	0.390	0.222	0.222	0.464	0.470	0.450	0.434	0.700
	DMM	**0.443**	**0.228**	**0.251**	**0.510**	**0.500**	**0.460**	**0.443**	**0.700**
	alone	0.342	0.217	0.205	0.482	0.462	0.436	0.424	0.687
	cSVM	0.342	0.219	0.224	0.480	0.466	0.440	0.425	0.682
CBA	cBST	0.384	0.154	0.195	0.461	0.453	0.432	0.426	0.693
	OR	0.376	0.209	0.200	**0.502**	0.456	0.437	0.426	0.688
	DMM	**0.400**	**0.223**	**0.235**	0.495	**0.477**	**0.450**	**0.436**	**0.698**
	alone	0.430	0.168	0.219	0.508	0.485	0.449	0.440	0.698
	cSVM	0.424	0.240	0.266	0.527	0.513	0.480	0.460	0.713
BST	cBST	0.440	0.171	0.229	0.481	0.457	0.440	0.430	0.690
	OR	0.438	0.239	0.261	0.551	0.525	0.491	0.470	0.723
	DMM	**0.450**	**0.243**	**0.280**	**0.560**	**0.535**	**0.495**	**0.475**	**0.730**
	alone	0.379	0.230	0.248	0.517	0.500	0.469	0.450	0.707
	cSVM	0.400	**0.240**	0.256	0.473	0.463	0.447	0.440	0.710
SVM	cBST	0.440	0.182	0.242	0.452	0.447	0.436	0.430	0.680
	OR	0.410	0.226	0.234	0.520	0.509	0.480	0.457	0.715
	DMM	**0.475**	0.235	**0.285**	**0.525**	**0.520**	**0.495**	**0.475**	**0.730**

Table 6.1. Annotation and retrieval results for CAL500. The performance is measured with a variety of metrics. The four first-stage auto-taggers from Section 4.3 (GMM, CBA, BST, SVM) are evaluated without context model (indicated as "alone"), in combination with 3 previously proposed, discriminative context models (cSVM, cBST, and OR), and in combination with our generative context model based on Dirichlet mixture models (DMM), respectively. The best results for each first-stage auto-tagger are indicated in bold.

and less for SMNs with more outspoken peaks (higher kurtosis); we use the kurtosis-based discrimination as reported in Section 4.4.1. Compared to using the same h for all training SMNs, this improves the F-score further to 0.251 and the MAP to 0.443.

6.3.2.2 Results on CAL500

Annotation and retrieval results for the CAL500 data set are presented in Table 6.1, using 5-fold cross-validation, annotations with the 10 most likely tags, and kurtosis-based SMN pre-processing for estimating DMMs as specified in Section 6.3.2.1. The results show that all first-stage (semantic) auto-taggers (GMM, CBA, BST, and SVM) benefit from adding DMM-based context modeling (compared to being used "alone", i.e. without context model), across

			Retrieval		
Category	# Tags	Model	P5	P10	MAP
Emotion	35	alone	0.513	0.506	0.477
		cSVM	0.539	0.510	0.480
		cBST	0.538	0.509	0.477
		OR	0.554	0.515	0.490
		DMM	**0.560**	**0.530**	**0.492**
Genre	11	alone	0.367	0.325	**0.355**
		cSVM	**0.392**	**0.336**	0.350
		cBST	0.363	0.330	0.344
		OR	0.360	0.316	0.339
		DMM	0.360	0.330	0.340
Instrument	14	alone	0.460	0.431	0.441
		cSVM	0.480	0.450	0.450
		cBST	0.460	0.444	0.442
		OR	0.490	0.450	0.448
		DMM	**0.495**	**0.458**	**0.460**
Usage	6	alone	0.253	0.233	0.258
		cSVM	0.250	0.220	0.235
		cBST	0.253	0.190	0.239
		OR	0.226	0.216	0.252
		DMM	**0.280**	**0.270**	**0.282**
Acoustic	25	alone	0.508	0.501	0.472
		cSVM	0.548	0.510	0.486
		cBST	0.524	0.503	0.477
		OR	0.561	0.524	0.500
		DMM	**0.566**	**0.533**	**0.500**
Vocals	6	alone	0.253	0.240	0.261
		cSVM	0.233	0.220	0.245
		cBST	0.273	0.246	0.263
		OR	0.280	**0.265**	0.270
		DMM	**0.287**	0.260	**0.275**

Table 6.2. Retrieval results per tag category, for CAL500. Semantic multinomials are computed by a GMM-based auto-tagger and evaluated without context model ("alone"), and in combination with a variety of context models (cSVM, cBST, OR, and DMM), respectively. The best results for each category are indicated in bold.

all performance metrics for both annotation and retrieval. For retrieval, context modeling often more clearly improves the precision-at-k metrics, which focus on the top of the ranked list of retrieval results. For the end user of a semantic music search engine, the quality of the top of the ranking is usually most important.

Most other approaches that leverage contextual tag correlations (i.e. cSVM and OR)

Rank	GMM-DMM	GMM
1	**Tim Buckley - Morning glory**	Yakshi - Chandra
2	**Charlie Rich - Behind closed doors**	Bread - If
3	Queen - We will rock you	**Charlie Rich - Behind closed doors**
4	**George Harrison - All things must pass**	Grateful Dead - High time
5	**Bonnie Tyler - Total eclipse of the heart**	Queen - We will rock you
6	Bread - If	Wes Montgomery - Bumpin'
7	Wes Montgomery - Bumpin'	Curandero - Aras
8	**Steely Dan - Rikki don't lose that number**	**Tim Buckley - Morning glory**
9	**Mazzy Star - Fade into you**	**Bonnie Tyler - Total eclipse of the heart**
10	**Carpenters - Rainy days and Mondays**	**George Harrison - All things must pass**

Table 6.3. Top-10 retrieved songs for "piano". Songs with piano are marked in bold.

improve the majority of the performance metrics, but often not as many as DMM does: the generative DMM approach is generally superior, irrespective of the underlying semantic model. cBST, on the other hand, frequently performs worse than all other context models and combining it with a first-stage auto-tagger often does not improve performance compared to not adding cBST.

In Table 6.2, we analyze the retrieval results per tag category (emotion, genre, instrument, etc.), for combining a GMM-based semantic model with a variety of context models (qualitatively, the results are similar for other semantic models). For categories of tags that regularly co-occur with other tags in the vocabulary (e.g. "Emotion", "Instrument", and "Acoustic", as can be seen in Figure 4.1), most context models improve the semantic model ("alone"), when combined with it. Combining semantic models with the DMM-based context model provides the best results, across all metrics. Categories of tags that co-occurr less frequently with other tags (e.g. the "Usage" and "Vocals" categories) still clearly benefit from DMM-based context modeling. Adding other context models provides mixed results and improves performance less than adding DMM. This indicates that even only a few co-occurrence patterns in the training data provide enough "detectable" information for the generative, DMM-based approach to improve auto-tagging performance, as opposed to other approaches which do not seem to pick up much contextual information. For the "Genre" category, co-occurrence patterns with other tags are not frequently observed in the CAL500 data set, as shown in Figure 4.1. For this category, adding any of the contextual models leads to mixed results while improvements, if any, are small. A single outlier is cSVM significantly improving P5. We have no good explanation for the latter.

Table 6.3 shows the top-10 retrieval results for the query "piano", both for GMM and the combination GMM-DMM. Lastly, Table 6.4 reports automatic 10-tag annotations for some songs from the CAL500 collection, with GMM and GMM-DMM.

	Jerry Lee Lewis "Great balls of fire"
GMM	calming, pleasant, folk, backing vocals, female lead vocals, **not angry**, not exciting, constant energy level, undanceable, **acoustic**
GMM-DMM	**lighthearted**, **emotional**, pleasant, **positive**, folk, **male lead vocals**, **not angry**, **like**, **acoustic**, vocal harmonies
	Alicia Keys "Fallin"
GMM	**emotional**, **tender**, **female lead vocals**, **piano**, acoustic guitar, slow, light beat, not recommended, **undanceable**, **vocal harmonies**
GMM-DMM	**emotional**, **pleasant**, **female lead vocals**, **piano**, **catchy**, **like**, **recommend**, **acoustic**, **emotional vocals**, **vocal harmonies**
	Lynyrd Skynyrd "Sweet home Alabama"
GMM	classic rock, rock, drum set, depressed, not happy, heavy, negative, negative feelings, not recommend, **undanceable**
GMM-DMM	alternative, classic rock, **country**, **piano**, drum set, **clean electric guitar**, **heavy**, not touching, **driving**, hanging with friends
	Bryan Adams "Cuts like a knife"
GMM	angry, alternative, hard rock, **rock**, distorted electric guitar, **not calming**, unpleasant, **negative**, **not likeable**, **negative feelings**
GMM-DMM	angry, alternative, hard rock, **rock**, **bass**, distorted electric guitar, **not calming**, **not mellow**, unpleasant, **heavy**
	Glenn Miller "In the mood"
GMM	romantic, sad, touching, soft rock, piano, light, negative, not likeable, negative feelings, undanceable
GMM-DMM	**emotional**, **positive**, soft rock, **jazz**, female lead vocals, piano, **catchy**, changing energy, recommend, high pitched

Table 6.4. Automatic 10-tag annotations for different songs. CAL500 ground truth annotations are marked in bold.

6.3.2.3 Results on Swat10k

To analyze the previous systems when dealing with weakly labeled data, we train them on the Swat10k dataset. Given its weakly labeled nature, this corpus is not well suited to evaluate performances. Indeed, since the absence of a tag does not imply it is not applicable to a song, true negative and false positive predictions cannot be reliably verified. Therefore, this experiment is conducted by training models on Swat10k and evaluating them, reliably, on the (strongly labeled) CAL500 dataset. The tags considered are the 55 tags that the Swat10k and the CAL500 corpus have in common (spanning 22 genres and 33 acoustic qualities). Because

Semantic Level	Context Level	**Annotation** F-Score	**Retrieval** P10	MAP	AROC
	Rand	0.117	0.149	0.192	0.523
	UpBnd	0.695	0.667	1.000	1.000
GMM	alone	0.217	0.325	0.352	0.729
	cSVM	0.185	0.319	0.349	0.718
	cBST	0.200	0.311	0.344	0.716
	OR	0.171	0.283	0.330	0.700
	DMM	**0.234**	**0.331**	**0.361**	**0.730**
CBA	alone	0.145	0.268	0.315	0.685
	cSVM	0.151	0.280	0.316	0.680
	cBST	0.155	0.278	0.314	0.679
	OR	0.130	0.245	0.290	0.650
	DMM	**0.180**	**0.287**	**0.324**	**0.689**
BST	alone	0.179	0.297	0.333	0.706
	cSVM	0.175	0.321	0.353	0.728
	cBST	0.152	0.292	0.325	0.697
	OR	0.150	0.275	0.318	0.670
	DMM	**0.195**	**0.325**	**0.359**	**0.732**
SVM	alone	0.200	0.300	0.355	0.720
	cSVM	0.190	0.311	0.356	0.725
	cBST	0.200	0.290	0.335	0.690
	OR	0.170	0.290	0.334	0.697
	DMM	**0.220**	**0.315**	**0.371**	**0.730**

Table 6.5. Annotation and retrieval results for training on Swat10k and evaluating performance on CAL500, for the 55 tags in common between both datasets. GMM, CBA, BST, and SVM are evaluated without context model ("alone"), and in combination with cSVM, cBST, OR, and DMM, respectively. The best results for each first-stage (semantic) auto-tagger are indicated in bold.

of the smaller size of the vocabulary, we automatically annotate each song with 5 (instead of 10) tags in the annotation task. Again, kurtosis-based SMN pre-processing is applied for estimating DMMs (as specified in Section 6.3.2.1). As mentioned in Section 6.1.2, audio clips are represented using Delta-ENT feature vectors for all experiments involving Swat10k, to accommodate copyright issues.

Annotation and retrieval results are reported in Table 6.5. The performance obtained after adding a DMM-based context model is better than for adding any of the other context models, for all metrics. In fact, none of the other approaches to modeling context is really convincing in this scenario: half or more of the performance metrics in Table 6.5 decreases when adding a cSVM, cBST, or OR context model to a first-stage auto-tagger. These three discriminative approaches are clearly suffering from the weak labeling of the training data. The generative DMM-based approach, on the other hand, improves annotation and retrieval performance

for all reported metrics and first-stage auto-taggers, compared to using the semantic level "alone", without DMM. Also, at the semantic level, it is observed that the GMM-based generative approach outperforms all other first-stage auto-taggers, for all annotation and retrieval metrics. Lastly, combining generative models at the semantic and the contextual level, the tandem GMM-DMM results in the best overall performance for training on this weakly labeled corpus.

6.4 HMM-based Retrieval Framework Experiments

In this section we report the evaluation carried out for the HMM-based retrieval framework described in Chapter 5. In particular, Sections 6.4.1 and 6.4.2 present the experimental setup and the results, while Section 6.4.3 reports some final considerations.

6.4.1 Experimental Setup

In the following paragraphs we introduce the evaluation tasks and the models included in the comparison.

6.4.1.1 Retrieval Tasks

The HMM-based framework is evaluated in the retrieval tasks introduced in Section 5.3.1, that is *query-by-description* and *query-by-example*.

In the *query-by-description* task, a combination of k distinct tags is provided as query and the songs are ranked by their relevance to them. The tags are taken from the finite size vocabulary $\mathcal{V}$; we consider queries composed by $k = 1, 2, 3$ tags. We believe that a real applicative scenario is very unlikely to have queries with more than three tags. The retrieval performances are measured by scrolling down the list and searching the rank of the songs human-annotated with the query-tags in the ground truth; metrics are averaged through all the queries considered.

In the *query-by-example* task, the query is a song and the retrieval function ranks the songs according to their similarity with it. The quality of the ranking list is measured with respect to the songs having the most similar semantic description with the seed in the human-based ground truth. This set is built by computing the KL divergence between the semantic descriptions of the query and of all the other items in the corpus, and by retaining the items showing the least divergence. In particular, for each song, we generally consider 30 and 50 relevant documents for CAL500 and Swat10k respectively (we assume that a larger collection may imply a larger number of relevant songs). Additionally, Section 6.4.2.2-(b) shows the results achieved using a different number of relevant documents in CAL500. Lastly, the purpose of the evaluation is to search the rank of the relevant songs in the ranking list, and to average the metrics over the size of the query set.

6.4.1.2 Compared Models

We compare the HMM-based framework with some alternative approaches, which use either tags or acoustic similarity alone, as well as their combinations. We implemented these strategies in order to have a number-based comparison using the same music representations. A more general comparison with another state-of-the-art model is provided in Section 6.4.3. In the experiments, we consider the following approaches:

- *TAG-based* (TAG): this carries out retrieval using the semantic descriptions alone [100, 101]. This model ranks the songs according to their KL divergence with the query-tags. In the query-by-description scenario, a k-tags query is mapped into a query multinomial $\boldsymbol{q} = \{q_1, ..., q_{|\mathcal{V}|}\}$, where $q_i = 1$ if the word w_i of $\mathcal{V}$ is in the query, and $q_i = \epsilon$ where $1 >> \epsilon > 0$ otherwise. This multinomial is then normalized, making its elements sum to one. In the query-by-example scenario $\boldsymbol{q}$ is the semantic multinomial of the seed song. Once we have a query multinomial, we rank all the songs in the database by their KL divergence with this $\boldsymbol{q}$;

- *Acoustic-based* (AB): this ranks the songs by their direct acoustic similarity with the query [60]. This model easily fits the query-by-example paradigm; conversely, it needs an extra-effort to be adapted to the query-by-description scenario. In fact, we need to define a song to be used as starting point for the acoustic similarity (i.e. we rank songs according to their similarity with this song). In these experiments, for each query-tag we consider as seed the song ranked first by the TAG model, and we rank the other songs by the acoustic similarity with it (note that the ranking lists resulting with the TAG and AB models achieve the same P1);

- *Weighted Linear Combination* (WLC): this combines acoustic and semantic similarities in a single measure through a weighted linear combination. In particular, if $\mathrm{TAG}(x, y)$ and $\mathrm{AB}(x, y)$ are the similarity values between the general songs x and y in the corresponding models, the WLC-based similarity is defined as:

$$\mathrm{WLC}(x, y) = 0.6 \cdot \mathrm{TAG}(x, y) + 0.4 \cdot \mathrm{AB}(x, y) \ . \tag{6.4}$$

  We chose the weighting coefficients that maximized the retrieval results[1]. Retrieval is carried out in the same way as the AB model, leading to the same considerations. Simple linear combination may also be performed on the ranks of TAG and AB models (i.e. use independently the two models to rank the songs and then combine the ranking lists). However, in preliminary experiments, we did not obtain good results and we prefer to discard this option;

[1] In preliminary experiments we tested other different simple ways of combining acoustic and semantic similarities, including max, min, median, and geometric mean. For the sake of brevity we do not report these preliminary experiments; yet, the weighted sum was the approach that achieved the best results.

- *Post-hoc Audio-based re-Ranking* (PAR): this incorporates audio similarity into an already existing ranking [50]. It was originally proposed to improve a tag-based music search engine that indexes songs based on related Web documents. In our case, the algorithm modifies the ranking achieved by the TAG model by also including the acoustic similarity[2]. Briefly, for each song x the PAR approach computes a new score that combines the tag-based rank of x, the tag-based rank of all the songs having x in their acoustic neighborhood, and the rank of x in all these neighborhoods. The songs are then sorted according to this new score. In the implementation, we followed the details reported in the referred paper.

Additionally, each experiment includes as baseline the random (Rand) and the upper bound (UpBnd) models introduced in Section 6.3.1; for both models, the semantic descriptions are used to retrieve songs in the same way as the TAG model.

6.4.2 Results

After discussing some preliminary results, this section provides the retrieval results on the CAL500 and Swat10k datasets. As of Section 5.4, CAL500 audio clips are represented using both delta-MFCCs and FPs, while Swat10k song representation relies on delta-ENTs only. In all the tables reported, the symbol (*) after a numeric value means that the difference with the corresponding second best measurement in that experiment is statistically significant ($p < 0.05$, paired t-test). The same statistical test is also implied when we talk about "*significant improvements*" in the text without providing supporting numbers.

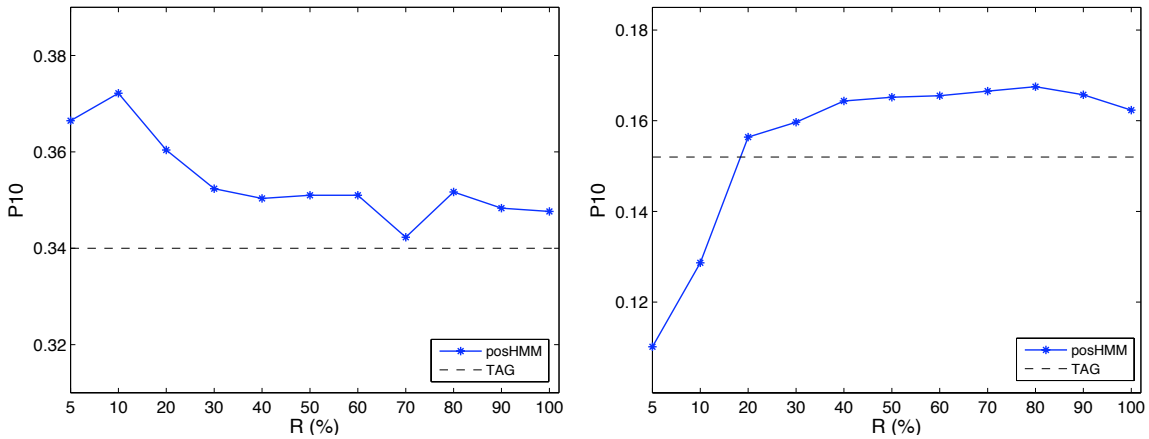
6.4.2.1 Preliminary Experiments

The HMM-based framework is a general model; however it is expected that some choices on the parameters affect the quality of the delivered ranking lists. In particular, the two possible tunings regard the connectivity of the model and the inclusions of the negative requirements. In this section we report the preliminary study carried out to define the best structure of the model; these results refer to the CAL500 dataset only.

(a) The Effects of the State Connections

Section 5.2.1 mentions the possibility of limiting the number of edges in the graph. The idea is to consider for each state only R outgoing transitions (i.e. towards the R most acoustically similar songs), in order to improve scalability (i.e. making the transitions matrix more sparse) and maximize retrieval performances. In this experiment, for each state we test the value of R from 5% (i.e. 25 connections) to 100% (i.e. 501 connections, that is all

[2] [50] also propose a mechanism that incorporates the acoustic similarity directly in the tag-based scoring scheme. However, this approach is less general and relies on information about the number of Web pages mined when associating a tag to a song. Additionally the results described in that paper for this methodology are not significantly different to the PAR ones. For these reasons, we did not include it in this study.



(a) **(b)**

Figure 6.2. The effects of the number of edges outgoing from a state (R expressed as percentage of the collection size) on retrieval with the positive HMM-based framework in terms of P10. (a) 1-tag query-by-description and (b) query-by-example. The dashed line represents the performance achieved by the TAG model.

Semantic	Model	P10	MRR
cb-auto-tag	posHMM	0.369	0.510
	negHMM	0.365	0.512
	cHMM	**0.382**	**0.556**

(a)

Semantic	Model	P10	MRR
Last.fm	posHMM	0.369	0.588
	negHMM	0.355	0.500
	cHMM	**0.390**	**0.620**

(b)

Table 6.6. The retrieval results for positive (posHMM), negative (negHMM), and combined (cHMM) retrieval in 1-tag query-by-description (149 queries, CAL500 dataset), in terms of P10 and MRR. (a) content-based auto-tags and (b) Last.fm tags.

the songs in the collection unless the state itself) of the collection size (i.e. the number N of states in the model), and we measure the P10 on the obtained ranking lists. Results are reported in Figure 6.2a for 1-tag query-by-description (149 queries, i.e. all the tags in the vocabulary), and Figure 6.2b for query-by-example (502 query-songs, i.e. all the songs in the collection). In this preliminary experiment we use only the content-based auto-tags and the positive ranking; the dashed line shows the performances of the TAG model which is considered as baseline.

As can be seen, the value of R strongly affects the retrieval performances. In the query-by-description, the model works better when R is small (i.e. R between 5% and 20% of N), meaning that a little connected model gives higher-quality results. We believe this may depend on the fact that a few connections produce more discriminative similarity transitions which better combine with the observations. Conversely, in the query-by-example task, the graph needs to be more connected to obtain satisfactory ranking lists (i.e. $R >= 40\%$ of N).

1-tag query-by-description - 149 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.214	0.185	0.186	0.174	0.161	0.358	0.179
	UpBnd	1.000	1.000	1.000	0.966	0.886	1.000	1.000
cb-auto-tag	TAG	0.295	0.315	0.334	0.340	0.336	0.463	0.304
	AB	0.295	0.366	0.350	0.328	0.295	0.484	0.262
	WLC	0.295	0.340	0.357	0.341	0.325	0.476	0.275
	PAR	0.375	0.393	0.380	0.365	0.346	0.529	0.310
	cHMM	**0.410***	**0.431***	**0.400***	**0.382**	**0.360**	**0.556***	**0.329**
Last.fm	TAG	0.375	0.396	0.390	0.365	0.340	0.537	0.270
	AB	0.375	0.324	0.302	0.297	0.266	0.530	0.242
	WLC	0.375	0.334	0.327	0.315	0.284	0.523	0.252
	PAR	0.416	0.402	0.395	0.369	0.348	0.572	**0.280**
	cHMM	**0.470***	**0.441***	**0.436***	**0.390**	**0.351**	**0.620***	0.278

Table 6.7. The retrieval results for 1-tag query-by-description with CAL500; we consider 149 queries, that is all the distinct tags in the vocabulary.

Following these results and considerations, from now on, we generally set $R = 0.1 \cdot N$ (i.e. 10%) for query-by-description, and $R = 0.6 \cdot N$ (i.e. 60%) for query-by-example.

(b) The Effects of the Negative Requirements

A possible extension to the model is the inclusion of the negative requirements as proposed in Section 5.3.2. This experiment investigates the benefits that this extension may lead to. In particular, we compare the retrieval results achieved with positive (posHMM), negative (negHMM), and combined (cHMM) retrieval in the 1-tag query-by-description (149 queries), in terms of P10 and MRR. Results are reported in Table 6.6 for both the semantic representations considered, content-based auto-tags (a) and Last.fm social tags (b).

As can be seen, while the negative ranking alone does not achieve very good results, its combination with the positive list leads to the best final results in both scenarios. Similar results not reported here for brevity have been achieved with the query-by-example task. Therefore, from now on, we consider only the combined framework (cHMM).

6.4.2.2 Results on CAL500

This section presents the retrieval results for the CAL500 dataset. The cHMM model refers to the parameters setting discussed in the previous section.

2-tag query-by-description - 3,684 cases				
Semantic	Model	P5	P10	MRR
	Rand	0.074	0.074	0.190
	UpBnd	1.000	1.000	1.000
cb-auto-tag	TAG	0.185	0.190	0.335
	AB	0.192	0.179	0.346
	WLC	0.193	0.193	0.339
	PAR	0.200	0.201	0.375
	cHMM	**0.227***	**0.230***	**0.390***
Last.fm	TAG	0.148	0.140	0.309
	AB	0.149	0.150	0.315
	WLC	0.163	0.161	0.322
	PAR	0.148	0.141	0.317
	cHMM	**0.200***	**0.187***	**0.379***

(a)

3-tag query-by-description - 3,000 cases				
Semantic	Model	P5	P10	MRR
	Rand	0.047	0.048	0.140
	UpBnd	1.000	1.000	1.000
cb-auto-tag	TAG	0.106	0.101	0.238
	AB	0.150	0.127	0.264
	WLC	0.131	0.135	0.260
	PAR	0.133	0.124	0.300
	cHMM	**0.166***	**0.177***	**0.338***
Last.fm	TAG	0.081	0.085	0.233
	AB	0.094	0.095	0.230
	WLC	0.102	0.098	0.240
	PAR	0.089	0.089	0.229
	cHMM	**0.123***	**0.110***	**0.284***

(b)

Table 6.8. The retrieval results for query-by-description using combinations of 2 (a) and 3 (b) tags over all the 149 tags of the CAL500 dataset. Each query has at least 10 relevant songs. For the 3-tag scenario, we sample a random subset of $3,000$ queries.

(a) Query-by-description

Retrieval results for query-by-description are reported in Table 6.7, 6.8a, and 6.8b, for queries composed of 1, 2, and 3 tags respectively. While the 1-tag task is performed over all the 149 tags of the vocabulary, in the other two cases we pre-filter the query sets by considering all the tag combinations having at least 10 relevant songs in the ground truth. This is mainly done for discarding combinations that associate contradictory descriptions (e.g. "bitter" - "sweet", "rock" - "classic music"). This leads to 3,684 distinct queries composed of 2 tags, and about 11,000 of 3 tags; in this last case, we retain a random sample of 3,000 queries, assuming them generally enough to evaluate the task.

As can be seen, the cHMM framework generally outperforms all the other algorithms over all the experiments, both with content-based auto-tags and Last.fm tags. The major benefits are at the top of the ranking list; in particular, the improvements in P1, P3, P5 and MRR are statistically significant compared to the PAR algorithm (which generally is the second better model), as well as to the TAG model. Conversely, retrieval along the full ranking list tends to decrease the effectiveness, as can be inferred in Table 6.7 by the minor improvement of the MAP. Nevertheless, we argue again that the most important aspect of a music ranking list is the quality at its top, which is the most interesting part for a user. For this reason Tables 6.8a and 6.8b report top ranking measures only; in these scenarios, cHMM works even better with significant improvements also for P10, showing a good robustness to combined queries.

Note that all the results based on Last.fm tags tend to show higher precision at the top and worse along the whole list with respect to the results obtained with the content-based

Rank	cHMM	TAG
	classic rock	
1	**The Cure - Just like heaven**	Electric Frankenstein - Teenage Shutdown
2	**Faith No More - Epic**	**The Cure - Just like heaven**
3	**The Adverts - Gary Gilmore's eyes**	Joy Division - Love will tear us apart
4	The Metallica - One	**The Adverts - Gary Gilmore's eyes**
5	Electric Frankenstein - Teenage Shutdown	**Boston - More than a feeling**
	trumpet	
1	**B.B. King - Sweet little angel**	Paul McCartney - Ebony and ivory
2	Carl Perkins - Matchbox	Carl Perkins - Matchbox
3	The Marvelettes - Please Mr. Postman	**B.B. King - Sweet little angel**
4	**The Doors - Touch me**	Captain Beefheart - Safe as milk
5	**Diana Ross - Where did our love go**	The Jefferson Airplane - Somebody to love
	driving	
1	Buzzcocks - Everybody's happy nowadays	Radiohead - Karma police
2	**Big Star - In the street**	Tom Waits - Time
3	**Bauhaus - Ziggy Stardust**	Drevo - Our watcher, show us the way
4	The Napoleon Blown Aparts - Higher Education	Buzzcocks - Everybody's Happy Nowadays
5	**Pearl Jam - Yellow Ledbetter**	The Napoleon Blown Aparts - Higher Education
	happy	
1	**Jackson 5 - ABC**	**Creedence CR - Travelin' band**
2	**Creedence CR - Travelin' band**	**Jackson 5 - ABC**
3	**White Stripes - Hotel Yorba**	Ray Charles - Hit the road Jack
4	Rolling Stones - Little by little	Louis Armstrong - Hotter than that
5	**XTC - Love at first sight**	ABC - Poison Arrow
	emotional	
1	**Alicia Keys - Fallin'**	Van Morrison - And it stoned me
2	**Shakira - The one**	Clarence Ashley - The house carpenter
3	**Chantal Kreviazuk - Surrounded**	Booker T and the MGS - Time is tight
4	**Evanescence - My immortal**	Stooges - Dirt
5	**Carpenters - Rainy days and Mondays**	**Alicia Keys - Fallin'**

Table 6.9. The top 5 songs in the ranking lists obtained by cHMM and TAG for 5 representative 1-tag queries from the CAL500 vocabulary. We refer to the content-based auto-tags experiments; relevant songs are listed in bold.

auto-tags. This depends on the fact that the Last.fm representation is rather sparse and noisy, since tags may also not have been assigned. When a tag is not assigned to a song, the retrieval algorithms rank that song randomly or just solely on the basis of acoustic similarity. This

query-by-example - 502 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.065	0.071	0.073	0.071	0.066	0.205	0.074
	UpBnd	1.000	1.000	1.000	0.966	0.886	1.000	1.000
	AB	0.177	0.165	0.159	0.144	0.131	0.340	0.121
cb-auto-tag	TAG	0.260	0.217	0.187	0.141	0.133	0.410	0.133
	WLC	0.256	0.205	0.185	0.141	0.136	0.402	0.133
	PAR	0.261	0.211	0.191	0.147	0.135	0.408	0.135
	cHMM	**0.285**	**0.227**	**0.204**	**0.172***	**0.150**	**0.450***	**0.144**
Last.fm	TAG	0.197	0.177	0.163	0.153	0.139	0.359	0.131
	WLC	0.203	0.185	0.177	0.155	0.143	0.368	0.132
	PAR	0.201	0.168	0.170	0.158	0.147	0.360	0.128
	cHMM	**0.279***	**0.220***	**0.196**	**0.180***	**0.159**	**0.439***	**0.145***

Table 6.10. The retrieval results in query-by-example with 502 queries over the CAL500 dataset.

leads to a less precise bottom part of the list, which affects the measurements. Conversely, tags generated through the auto-taggers are more stable since each song has a relevance value for each tag, and no song is retrieved randomly or according to acoustic similarity alone.

A deeper analysis of the results in the 1-tag query-by-description showed that the improvement in P10 with respect to the TAG model involves 121 and 119 tags for the content-based auto-tags and the Last.fm tags respectively (about 80% of the queries); conversely, the PAR algorithm improves only about 71% of the queries. The improvement generally happens in all the tag categories, meaning that there are no types of tags that gain a major benefit (CAL500 spans different categories of tags, such as genre, emotion, usage, etc.). A similar result is achieved with 2-tag queries, while the improvement in the 3-tag scenario is even greater with about 87% of the queries outperforming the TAG model in terms of P10.

Lastly, Table 6.9 compares the top five positions in the ranking list achieved by TAG and cHMM for 5 representative 1-tag queries (content-based auto-tags). This table aims at showing some examples of the results delivered to a user of a semantic music discovery engine, especially highlighting the benefits introduced by the cHMM framework. For example, in response to the query "emotional", cHMM ranks 5 relevant songs at the top, compared with only one ranked by the TAG model.

(b) Query-by-example

The query-by-example results for CAL500 are presented in Table 6.10. We use all the songs of the dataset as query; therefore the results are averaged over 502 results. We consider 30 relevant songs for each query, which have been gathered as described in Section 6.4.1.1.

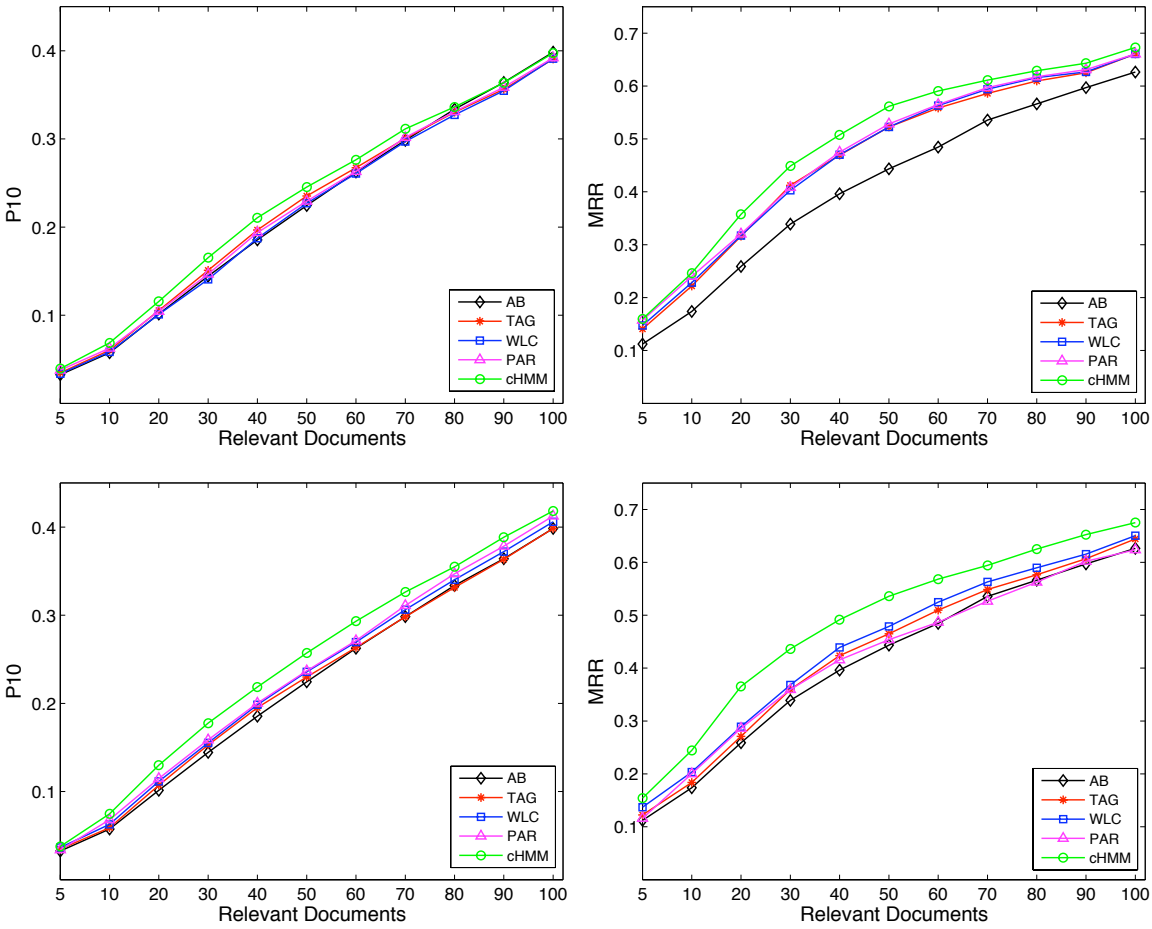


Figure 6.3. Retrieval results in query-by-example for different numbers of relevant songs in terms of P10 and MRR; (a),(b) content-based auto-tags, (c),(d) Last.fm tags.

As can be seen, also in this scenario cHMM generally outperforms all the other algorithms. Major improvements are achieved with the Last.fm tags (significant for P1, P3, P10, and MRR). These results show that combining through cHMM completely different sources of information (in this case social tags and music content) may be a competitive strategy. The improvement of P5 is not significant at the 5% level; however, the t-test obtains $p = 0.064$, then very close to be significant. Conversely, results with the content-based auto-tags show significant improvement for the P10 and MRR only.

It should be noted that this evaluation may be slightly affected by the algorithm used to choose the relevant documents. In fact, for each query we automatically choose in a heuristic way the first 30 most similar songs according to the human-based annotations. Nevertheless, some queries could not have 30 "true" similar songs. For example, the tag "swing" has only 5 human-annotated examples in the ground truth; this means that these songs could not have up to 30 songs which can be truly considered similar in the dataset (the "swing" is a quite strong discriminative music genre). Therefore, the evaluation could have been done by also searching songs that are not truly relevant (i.e. these songs are in the top-30 only because

1-tag query-by-description - 485 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.029	0.028	0.023	0.022	0.021	0.075	0.023
	UpBnd	1.000	1.000	1.000	1.000	0.985	1.000	1.000
	TAG	0.212	0.200	0.216	0.226	0.227	0.361	0.145
	AB	0.212	0.254	0.246	0.211	0.189	0.390	0.085
cb-auto-tag	WLC	0.212	0.241	0.243	0.221	0.192	0.372	0.117
	PAR	0.278	0.262	0.256	0.236	0.233	0.434	0.106
	cHMM	**0.315**∗	**0.293**∗	**0.281**	**0.270**∗	**0.259**	**0.463**	**0.163**

Table 6.11. The retrieval results for the 1-tag query-by-description with the 485 tags of the Swat10k dataset.

query-by-example - 1,000 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.004	0.004	0.004	0.005	0.005	0.025	0.006
	UpBnd	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	TAG	0.121	0.109	0.103	0.090	0.078	0.233	0.039
	AB	0.113	0.098	0.091	0.084	0.076	0.222	0.047
cb-auto-tag	WLC	0.164	0.134	0.119	0.104	0.089	0.279	0.052
	PAR	0.169	0.139	0.122	0.106	**0.092**	0.285	0.052
	cHMM	**0.214**∗	**0.162**∗	**0.145**∗	**0.122**	0.089	**0.361**∗	**0.058**

Table 6.12. The retrieval results in query-by-example with $1,000$ random queries in the Swat10k dataset.

less *dissimilar* to the query than the others).

For this reason we also provide Figure 6.3, which depicts the performances of the different models in terms of P10 and MRR by ranging the automatic number of relevant songs from 5 to 100. As can be seen, cHMM generally outperforms all the other models in any case. Note that when the number of relevant documents is higher, all the models tend to converge to the same performance.

6.4.2.3 Results on Swat10k

This section presents the results with the Swat10k database. The purpose of the following experiment is two-fold. On the one hand, we aim at testing the framework with a larger collection, both in terms of tags and songs. On the other hand, the test involves also the music content descriptors; in fact, as mentioned in Section 6.1.2, the Swat10k songs must be

represented using Delta-ENTs to accommodate copyright issues.

From the original semantic vocabulary we retain only the tags with at least 10 examples in the ground truth in order to have a reasonable minimum number of songs to search during the evaluation. The final vocabulary is then composed of 485 distinct tags (119 "genre" and 366 "acoustic" tags). Note that no subjective tags such as emotions or usages are included in this database. With this dataset, we use only the content-based auto-tags for semantically describing each song in order to have more dense descriptions. In fact, the Pandora-based Swat10k tags are assigned to songs by experts and musicologists, and sometimes prove too specific and technical to be also given by normal Last.fm users (e.g. "slide/pedal steel guitars", "electric guitar wall-o-sound", "dominant melodic hooks", etc.). Therefore, the resulting Last.fm-based descriptions would be too sparse to deliver reliable ranking lists (i.e. where a few random-based sorts are involved) for evaluation purposes.

During preliminary experiments not reported here for brevity we saw that the framework built using all the songs of the database leads to very low retrieval results. We believe this may depend on the normalization performed on the transition probabilities to satisfy the stochastic requirements of the HMMs theory. In fact, with many edges outgoing from each state (e.g. about $6,100$ for query-by-example), the normalization leads to similar transition values which do not discriminate well the paths across the model.

Nevertheless, we argue that in the real scenario of commercial search engines, it would be computationally complex to build the graph with all the millions of songs indexed as well. For this reason, we propose using the cHMM framework to refine the retrieval on a reliable subset of the collection (i.e. reliable in the sense that ideally all the relevant songs are in this subset). One method to extract this subset is to pre-cluster the songs using metadata such as title, artist, year, user preferences, etc. Since we search by tags and we do not have many other additional pieces of information to exploit in this scenario (e.g. artists similarity), we use the TAG model as clustering algorithm. In particular, for each query, we consider the top 2,000 results achieved by the TAG ranking and we use cHMM to re-rank this subset. We measured that the recall-at-$2,000$ (i.e. the number of relevant documents retrieved over the total number of relevant documents) achieved by the TAG model is about 70% for both tasks; therefore the resulting clusters can be considered reliable enough to be used as input for the cHMM algorithm.

We use cHMM with the same parameters guideline defined in Section 6.4.2.1. The query-by-description task is performed over 485 1-tag queries (i.e. all the tags of the vocabulary); results are reported in Table 6.11. We did not investigate the 2-tag and 3-tag queries scenario with this dataset. As can be seen, the cHMM framework generally leads to the best results again. First, we obtained a significant improvement with respect to TAG and AB in several metrics; additionally, results are significantly better on P1 and P10, and never worse than the PAR model (which generally is the second best model at the top of the ranking lists). It can also be noted that AB works better than with CAL500; this is due to the more qualitative timbre descriptions provided by the Delta-ENT features.

Lastly, Table 6.12 shows results for query-by-example; in this experiment we randomly sampled 1,000 songs from the dataset to use as seed queries. Again, cHMM outperforms the other models with significant improvements at the top of the ranking list.

6.4.3 Discussion

The experimental evaluation carried out to validate the cHHM framework shows that the proposed model generally outperforms other state-of-the-art approaches that rank songs by a single source of information alone, or by alternative their combinations. To the best of our knowledge, the models included in the comparison represent the state-of-the-art concerning systems that combines acoustic similarity and tags for music search and discovery. Alternative approaches in the literature use other sources of information, such as artist and user related data, or Web documents, and combine them for other retrieval tasks (e.g. artist clustering and recommendation, classification). However, a comparison with these methodologies goes beyond the scope of this work. We believe that the high generality of the model makes it possible to integrate such different descriptions as well.

Nevertheless, one additional approach that should be mentioned, although it does not exploit audio similarity when combining different sources of information, is [99]. In this work, the authors combine semantic descriptions for a music discovery task. In particular, they consider content-based auto-tags, social tags and tags mined from Web pages, while the combination is carried out through three different machine learning approaches. Experiments on 72 tags of CAL500, one of the collections used in our experiments, show that the combination approach called Calibrated Score Averaging outperforms the others (including the use of single descriptors) in terms of MAP. When examining each tag individually, they report that the percentage of tags that are improved by each combination model with respect to the best retrieval achieved using individual descriptors ranged from 15% to 27%. Conversely, the percentage of tags improved by at least one combination approach was 65%. While a global direct comparison based on the absolute value of the measures cannot be done because of the smaller vocabulary and of the shorter ranking lists due to the cross validation procedure, we can compare improvements when tags are taken individually. So, we detect that for experiments in query-by-description using a single tag, cHMM improves MAP for 68% of the content-based auto-tags and 60% of the Last.fm tags. Therefore, we expect a greater improvement in the retrieval results also with respect to the combination of different semantic sources, although this may depend on the choice of the descriptors used in the combination.

CHAPTER 7

Conclusions

During the last years, the dynamics of the music industry have evolved to face all the changes imposed by the rise of the Internet. In particular, stores of physical media have been largely replaced by online services where it is possible to buy, download and listen to music content. Yet, all this music proliferation has highlighted a need for automatic music technologies that allow the users to interact with these large music corpora easily and effectively. A valid strategy is to process the information carried by the audio signal to deliver fully functional and automatic content-based solutions.

Along this thesis, we have addressed several issues that concern content-based music information retrieval. In particular, we have mostly focused on automatic music tagging, music discovery and music identification systems. In this chapter, we dedicate some lines to give a brief summary of the original achievements and a general discussion of the content-based approaches and their implications in commercial systems. Finally, an overview of possible future directions closes the chapter.

7.1 Overview of the Original Contributions

The research in this thesis can be described as interdisciplinary because it involves computer audition, digital signal processing, machine learning, information retrieval, natural language processing, music theory, and musicology. After an introduction and a review of the music information retrieval research area, the thesis has reported some original contributions and achievements.

First, we have proposed a novel, generative approach to modeling contextual relationships between tags to improve automatic music tagging. In particular, for each tag, we estimate a Dirichlet mixture model to capture typical tag co-occurrence patterns (i.e. "context") in

semantic multinomials that songs associated with that tag have been annotated with. This results in a two-step approach: in a first stage, an existing auto-tagging system that allows to annotate a song with a semantic multinomial is applied; in a second stage, the resulting semantic multinomial is refined by a stage of Dirichlet mixture models, based on contextual evidence. This results in a set of posterior tag probabilities that provides a contextual description of the song, i.e. a contextual multinomial. The generative character of this approach makes it well suited for weakly labeled data sets and naturally allows us to rank tags probabilistically for a song. We have shown empirically that modeling context with DMMs improves performances when combined with a variety of first-stage (semantic) auto-taggers, compared to using the first-stage auto-taggers alone. Moreover, the generative DMM-based approach generally outperforms other, discriminative approaches to modeling context. Its superiority is more outspoken when trained on weakly labeled data. Examining the performance per tag category reveals that DMM-based context modeling most significantly benefits those categories of tags that have been empirically observed to frequently co-occur with other tags.

In order to reduce the limitations of the retrieval based on auto-tags alone, we have also proposed a novel approach for music retrieval based on a hidden Markov model that combines acoustic similarity and tags in a single statistical framework. Each state of the model represents a song of the music corpora, transitions between states are ruled by acoustic similarity, while tags define the state observations. An algorithm based on Viterbi decoding is used to retrieve the list of songs which best respond to a user query, which is expressed either as a combination of tags or as a seed song. Additionally, the model exploits the negative requirements which are latent in the query for improving the quality of the ranking list. In the presence of tagged music corpora, the framework does not require any additional preprocessing steps (e.g. training); additionally, efficiency is guaranteed by sub-retrieval steps which reduce the waiting time of the user. This approach aims at integrating and improving state-of-the-art systems for semantic music search and discovery engines, recommendation and playlist generation. Experimental evaluation shows that the proposed model generally outperforms other state-of-the-art approaches that rank songs by a single source of information alone, or by a combinations of information sources. Improvement is more significant at the top of the ranking lists and when the size of the collection is smaller. In contrast, a larger collection brought up some performance issues; however, the model is particularly suitable for re-ranking a cluster of relevant candidates efficiently retrieved from a collection.

Lastly, we have reviewed a classical music identification system that relies on the two original music descriptors presented in the appendixes and summarizing the harmonic content of the music signal. The system is based on two steps. At first it extracts a subset of relevant songs for a given query in order to select a cluster of potential candidates from the collection. The second step computes the similarity between the audio query and each recording in this subset through an application of HMMs. Therefore identification is carried out using two different strategies, the first to achieve efficiency, the second to refine effectiveness. The

experimental results show how the system achieves satisfactory results both in terms of efficacy and efficiency over a collection of about 13,000 different music works.

7.2 Discussion

The use of content-based approaches has always been a common strategy in the music information retrieval area. However, despite the large amount of research carried out in the last decade on the matter, the problem is far from being solved and there is still space for new solutions. For this reason, commercial music retrieval systems tend to not use content-based approaches, but they generally rely on metadata and social tags only. The most famous application of MIR content-based strategies in a commercial scenario is probably Shazam!, which relies on audio fingerprinting elaborations. Nevertheless, it should be noted that to process the audio signal is the only way of dealing with unknown recordings where both tags and metadata are unavailable. In contrast, commercial music search and discovery systems where tags and metadata are generally accessible, do not use content-based techniques in the ranking process.

However, nowadays research systems on music retrieval may be ready to compete with commercial solutions. In fact, in an informal study of music recommendation systems, Lamere and Celma found that an academic prototype based on semantic similarity outperformed 12 commercial and academic systems based on social similarity (i.e. collaborative filtering) and five independent human experts [55]. In addition, more recently Barrington et al. [6] show that a recommender system built on content-based auto-tags (i.e. GMM model) can expect to perform similarly or even better than Apple iTunes Genius when exploring the long tail. Genius is based on collaborative filtering of huge amounts of social data; however, in the case of less known music this massive quantity of data is partially, or even totally, missing and Genius is unable to make good recommendations. In contrast, Genius works better with popular music, that is famous songs and artists.

These considerations as well as the results reported in the thesis provide a strong justification for the inclusion of content-based approaches in the commercial music systems of next generations. In fact, content-based solutions can now effectively help to explore the long tail of less known and recommended songs/artists; the purpose is to reach a *market equilibrium* where all the songs have fairly reasonable chances to be included in the recommended lists, and therefore to be promoted in the market. Additionally, content-based solutions allow the systems to drastically reduce the waiting time for new released songs to be catalogued, tagged, and indexed; in fact, all these operations can be done automatically and immediately without relying on human-efforts with still satisfactory results in terms of efficacy.

Nevertheless, while state-of-the-art content-based strategies can now provide efficient and effective solutions, we argue that they need to be combined with social data in any case in order to fully encounter the user satisfaction. In fact, content-based processing leads to very *objective* representations of the audio signal and similarity relationships; however, music

perception is highly *subjective* and change from person to person. This subjectivity is not considered when using fully content-based solutions. Additionally, in the case of music discovery systems, the available commercial solutions built on social data are currently proving to satisfy the users; in fact, as of Chapter 1, the number of users using these systems is rapidly increasing, especially because of the good results with popular music. For all these reasons, we believe that content-based strategies should aim to integrate the social data for obtaining final systems which provide the best recommendations. In fact, these hybrid solutions would treat equally and efficiently all the documents in the collections because of the automatic content-based processing and, at the same time they would keep into account the subjectivity of the human music perception as well. This consideration is also partially motivated by the results obtained combining social tags and acoustic similarity through the cHMM framework, which show the large improvements in retrieval given by adding content-based processing (i.e. acoustic similarity) with respect to rank songs by social tags-only.

Similarly, also music identification in a scenario of audio cataloguing will always require a final interaction with a human user. In fact, a fully unsupervised identification task could be feasible only in presence of a system which guarantees to rank a correct document always at the first place; however, we argue that such system is quite unlikely to be built because of the infinite variables involved in the process (e.g. noise, errors in the playing, personalizations of the artist). For this reason, a user that checks the validity of the final result is necessary in order to ensure a correct cataloguing of the audio recordings.

7.3 Directions for Future Works

Several improvements and considerations regarding the retrieval models exposed in this thesis can be done. In this section we briefly review some of them as well as some more general directions for future works in MIR.

As regards the DMM, we proposed this model for improving auto-taggers by modeling the contextual relationships between tags. However, modeling patterns at the SMN level has the advantage of working with a representation that is independent of the low-level audio feature representation and the actual semantic auto-tagging model used. Even more, SMNs obtained from different auto-taggers or with different audio features may be concatenated, re-normalized, and modeled, once again, as a sample from a (larger) Dirichlet mixture model. Future works may explore this approach as a way to integrate predictions from different auto-taggers or based on different audio features, at different time resolutions. At the same way, the DMM can also be used to combine semantic descriptions related to different information sources (e.g. content-based auto-tags with social tags and/or Web mined tags).

As regards the HMM-based retrieval framework, future works may concern the integration with collaborative filtering techniques for leading to the hybrid solution mentioned in the previous section. This can be done exploiting user-related data in the definition of transition probabilities to take into account the subjectivity of music similarity, thus without modifying

the retrieval algorithm. Alternatively, the HMM may be extended to a more complicated structure, such as hypergraph or hierarchical HMM; however, this needs to extend the retrieval function as well. In addition, the high generality of the model makes it suitable for other media; in particular, it would be interesting to explore its applicability to an image retrieval task.

In addition, we believe that future research in music discovery systems should investigate the scenario of time-varying queries. An applicative example is a person creating a music playlist for a party; he may prefer mellow and quiet music at the beginning, danceable and fast tempo music at the middle, and quiet music again at the end of the party. We believe that retrieval model should deal automatically and effectively with this scenario; at this aim, it would be interesting to test how state-of-the-art approaches work, and in case to propose novel ad-hoc solutions.

Lastly, nowadays MIR algorithms are set to work in functions of features and parameters, which are trained and tested for obtaining the results in terms of precision, recall, or whatever measure. After training, all these variables always do their duty in the different task they are involved, generally providing the user with ranking lists that should maximize the number of relevant documents in the first positions. Nevertheless, after a while such results can be considered boring, or at least less interesting. In contrast, it happens frequently to a person that a song is unexpectedly nice, even if it is not of the right genre, played with the right orchestration, and so on. In this latter case, some of the parameters on which a user bases his judgement become irrelevant, or even assume a negative weight. Future works within the MIR community should consider these scenarios for automatic music discovery as well (i.e. modeling novelty in music retrieval); we believe that the application of game theory techniques to MIR could be a valid and original starting point.

APPENDIX A

Rank-based Chroma Hashes

Chroma features are common content descriptors which have been extensively used in a number of MIR applications. The concept behind chroma is that octaves play a fundamental role in music perception and composition [7]. For instance, the perceived quality – e.g. major, minor, diminished, etc. – of a given chord depends only marginally on the actual octaves where it spans, while it is strictly related to the pitch classes of its notes. Following this assumption, chroma features can well generalize the harmonic content of the audio signal [46]; a number of techniques have been proposed based on chroma features, including chord estimation [77], extraction of repeating patterns in pop songs [42], and music identification [15, 53, 69, 85].

As presented in Section 2.2.6.1-(a), we propose to use chroma features as pointers to the recordings they belong to, playing the same role of words in textual documents; therefore, each song results described by a sequence of *chroma words* [69]. In the following, we report *how to map a chroma vector into a chroma word*; all the main components of this process are depicted in Figure A.1.

A chroma vector $\bar{c}$ is a 12-dimensional array of pitch classes computed from the Fourier transform $\hat{X}$ of an audio frame X (i.e. half-overlapped windows of 23 ms extracted from the audio signal). Several methodologies are available in literature to compute chroma features. Among all, we follow the approach proposed by Ellis and Poliner [35], which uses the instantaneous frequency within each FFT bin of a frame in order to identify strong tonal components and to achieve higher resolutions. An example of chroma vector is shown in Figure A.2; the x-axis values correspond to the pitches *C*, *C#*, *D*, *D#*, *E*, and so on, whereas the y-axis values refer to the energy measured at each pitch class.

Although chroma vectors are already a compact representation of a music signal since they subsume the harmonic content of each audio frame in only 12 pitch classes, we propose

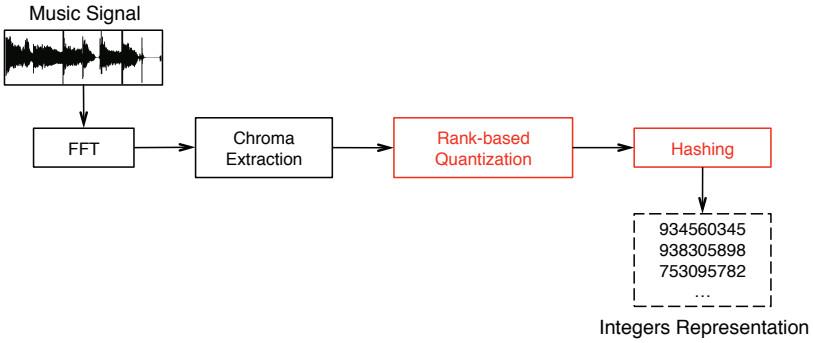


Figure A.1. General block diagram to represent music content as a sequence of chroma hashed integers.

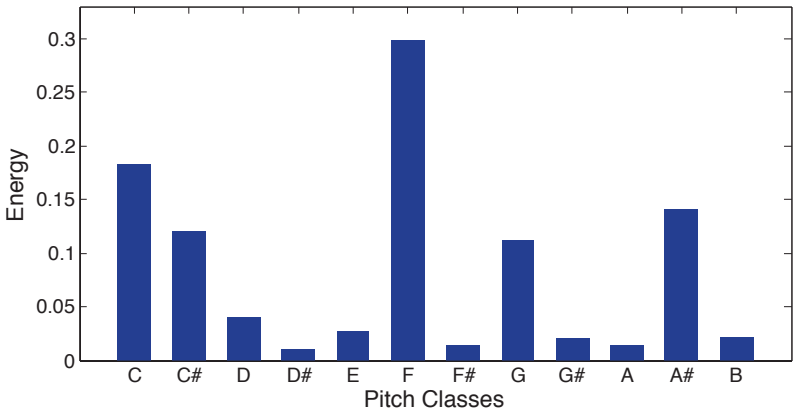


Figure A.2. A 12-bin chroma feature vector example; the x-axis values correspond to pitches *C*, *C#*, *D*, *D#*, *E*, and so on, whereas the y-axis values refer to the energy measured at each pitch.

to simplify this representation even more by applying an hashing technique to describe each frame as a single integer value. The main purpose is to reduce the computational cost of content-based music retrieval functions, because a similarity search in a 12-dimensional space is substituted by a comparison between integers.

As shown in Figure A.1, the approach is based on an hashing function applied to a quantized representation of the chroma vector. We propose to perform quantization following the basic idea that, within a vector, the order of the pitch classes according to their energy is more relevant than the actual value of the energy; therefore, quantization is carried out on the rankings of the pitch classes, sorted in decreasing order. In contrast, Kurth and Mueller [53] use the energy values to quantize the chroma vectors; however, preliminary experiments in the task of classical music identification showed that the rank-based quantization lead to better results than using such energy-based approach.

In addition, it is expected that the first elements in the ranking list will be related to the dominant frequencies, and then to the main harmonic content of the signal; conversely,

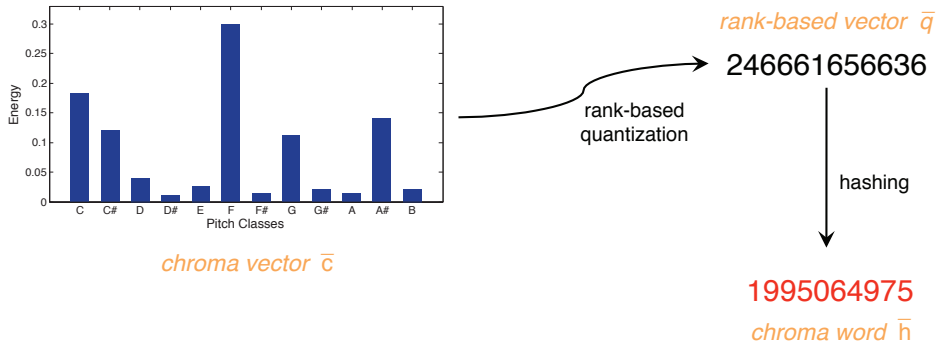


Figure A.3. An example of the rank-based hashing process: the chroma vector is first quantized, and then hashed to obtain the final integer value (i.e. chroma word).

the last elements will have values close to 0 and their ordering in the ranking list will be almost random. For this reason, the rank-based representation $\bar{q}$ of the chroma vector $\bar{c}$ can be computed using the ranking operator $rank_i(\cdot)$, which outputs the position of the i^{th} pitch class along the energy-based ranking list (i.e. $rank_i(\bar{c}) = 1$ if the pitch class i has the highest energy, 2 if is ranked as second, and so on) as follows:

$$\bar{q}(i) = \begin{cases} rank_i(\bar{c}) & \text{if } \ rank_i(\bar{c}) < \Upsilon \\ \Upsilon & \text{elsewhere,} \end{cases} \tag{A.1}$$

where $i \in \{1 \dots 12\}$ and Υ represents the number of quantization levels. Previous experiments for classical music identification using a small collection of 500 recordings set $\Upsilon = 6$ as optimal value [69].

The final compact representation is obtained considering that vector $\bar{q}$ can be thought as a twelve digit number represented in base Υ. Therefore, a simple hashing function can be computed by obtaining the decimal representation $\bar{h}$ of this number; in particular:

$$\bar{h} = \sum_{i=1}^{12} \Upsilon^{i-1}(\bar{q}(i) - 1) \ , \tag{A.2}$$

where the resulting $\bar{h}$ is the final *chroma word*. Additionally, it is possible to apply a second hashing technique for storing $\bar{h}$ in one array that can be accessed in constant time. A typical approach is to compute the remainder of $\bar{h}$ divided by a carefully chosen prime number [26]. Figure A.3 shows the described hashing process performed using the chroma vector depicted in Figure A.1.

Lastly, it is well known that chroma representations are sensitive to transpositions, that is the same audio excerpt played in two different tonalities is characterized by two different chroma representations (see Section 2.2.4). This may have a terrible impact on a music retrieval system: in fact, if the query and its matching versions in the collection are played in different tonalities, they will have totally different chroma descriptions that can be difficultly matched by the retrieval function. This problem can be addressed by considering that a

transposition of s semitones corresponds to a rotation of the vector $\bar{c}$ of s steps. Therefore, the value $\bar{h}_s$ of a chroma word $\bar{h}$ transposed by s semitones can be computed as follow:

$$\bar{h}_s = \Upsilon^{12-s} \cdot (\bar{h} \ \% \ \Upsilon^s) + \left\lfloor \frac{\bar{h}}{\Upsilon^s} \right\rfloor , \tag{A.3}$$

where $s \in \{1 \dots 11\}$, % gives the reminder of a division, and $\lfloor \cdot \rfloor$ is the floor function. Hence, in a retrieval task, a query can be represented by twelve sets of chroma words $\bar{H}_s$, that is the representation $\bar{H}_0$ in the original tonality and the eleven additional ones that take into account all the possible transpositions; this representation can be considered as an application of the *query expansion* typical of textual retrieval systems [62] (see Section 2.2.6.1-(a)).

APPENDIX B

A Binary Representation of the Spectral Content

Although it is particularly suitable for describing the content of a music signal, the frequency representation is costly in terms of space. For example, the 32-bits floating point representation of a windowed signal of 4096 samples, which is a typical value in music processing applications and which corresponds to about 93 milliseconds for a signal sampled at 44.1 KHz, requires 8 KB of space in memory if all the bins up to the Nyquist frequency are considered. We propose to represent the frequency content in binary form, that is every *bin* is represented by a *single bit*. This approach reduces the amount of memory required to represent the signal by an order of magnitude; indeed, using the values of the previous examples, the same 93 milliseconds can be memorized in 256 bytes.

The basic idea is to compute the values of the bits in the representation depending on their contribution to the overall energy of the signal. This approach is particularly suitable for representing an audio signal, where pitch information is usually one of the most relevant music feature and is related to the presence of peaks in the spectrum of the signal. In particular, the features extraction step is based on the computation of local maxima in the Fourier transform of the signal, which are likely to be related to the first harmonics of the events that are played in each frame. This approach can be applied to individual frames of an audio signal, as described above, as well as to a cluster of frames sharing a similar spectral content. To this aim, segmentation can be used to pre-process the music signal in order to have a single descriptor that summarizes consecutive audio frames with a similar frequency content. This allows for a substantial decrease in space requirements. Figure B.1 depicts the flow of activities to extract the proposed descriptors.

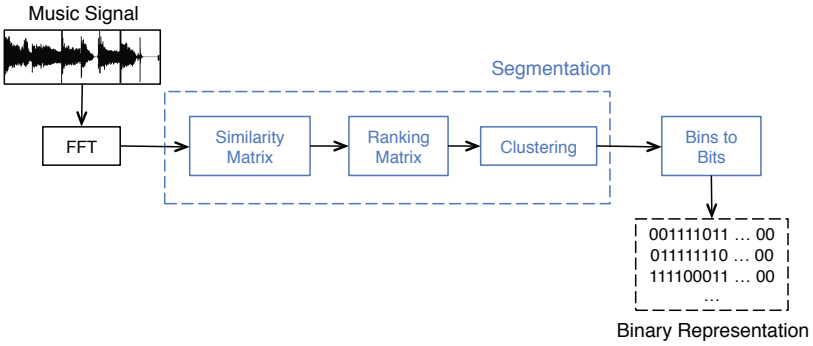


Figure B.1. General block diagram for computing the binary descriptors, from the audio content extraction to the binary strings conversion.

B.1 A Clustering Approach to Music Segmentation

Segmentation is used to group together consecutive frames that share a similar spectral content. Boundaries between two segments are related to changes in the spectral content, which in Western music usually depend on notes or percussive sounds being played or stopped. The use of spectral differentiation has been proposed for onset detection by Gouyon et al. [44], where a measure of novelty in the frequency representation is used to highlight note onsets. In our case, local differences in spectrum are exploited in a clustering process, following the approach proposed by Choi [25] for text segmentation.

The audio signal of a music recording is divided in half-overlapped frames of L samples, and the frequency representation is computed. The first step of the algorithm is the computation of the similarity among these frames, which is carried out as the cosine of the angle between them. In particular, given $\hat{X}$ and $\hat{Y}$ as column vectors containing the magnitude of the Fourier transforms of frames X and Y respectively, the similarity is defined as:

$$\mathrm{sim}(X,Y) = \frac{\hat{X}^T\hat{Y}}{|\hat{X}|\cdot|\hat{Y}|} \ . \tag{B.1}$$

High similarity values are expected for frames where the same music events – i.e. notes, percussive sounds, etc. – are played. Conversely, a drop in the similarity between two subsequent frames is related to a change in these music events. A similarity matrix $\bar{\boldsymbol{S}}$ can be defined as $\bar{\boldsymbol{S}}_{XY} = sim(X,Y)$; an example of the similarity matrix achieved for a short music excerpt is shown in Figure B.2a. High similarity values are represented by bright pixels; the diagonal from top-left to bottom-right shows the self-similarity of the consecutive frames. The square regions along the diagonal represent the potential similar frames that should be clustered together.

However, the pure similarity values may not be completely reliable for a segmentation task because changes in the local correlation could be more relevant than absolute similarity values. For instance, the value of a local correlation of a note sung with vibrato is expected to be lower than in the case of a steady tone played by an organ. Nevertheless, in both

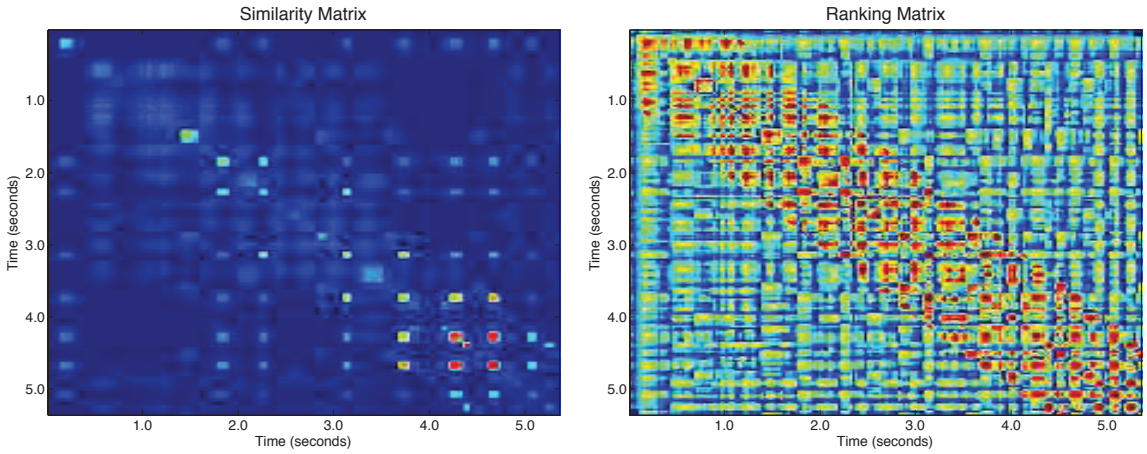


Figure B.2. Similarity matrix (a) and ranking matrix (b) of an excerpt of about 6 seconds of music.

cases a decrease in local correlation is expected for changes in the active music events. For this reason, segmentation is carried out on a *ranking* matrix. In non-parametric statistical analysis the rank of data sets are generally compared when qualitative behavior is similar but the absolute quantities are unreliable; a similar approach can be also extended to the segmentation process. Therefore, for each couple (X, Y) of frames in $\bar{\boldsymbol{S}}$, the similarity value is substituted by its *rank*, which is defined as the number of neighbor elements whose similarity is smaller than $sim(X, Y)$. That is, the elements of the ranking matrix $\bar{\boldsymbol{R}}$ are computed as

$$\bar{r}_{XY} = ||(A, B) : sim(A, B) < sim(X, Y)|| \quad \text{with} \quad (A, B) \in N(X, Y) \ , \tag{B.2}$$

where $N(X, Y)$ denotes the set of neighbors of the entry (X, Y) in $\bar{\boldsymbol{S}}$. In our experiments, $N(X, Y)$ corresponds to a square centered in (X, Y) with a length of 10 frames. $N(X, Y)$ is suitably reduced when $\bar{r}_{XY}$ is computed at the borders of the similarity matrix. Figure B.2b shows the ranking matrix achieved by processing the similarity matrix of Figure B.2a; as can be noted, similar parts are more easily identified in this representation.

Once the rank is computed for each couple of frames, the ranking matrix $\bar{\boldsymbol{R}}$ is processed through hierarchical clustering, which attempts to segment the music signal in coherent passages. The clustering step computes the location of boundaries using Reynar's maximization algorithm [84], a method for finding the segmentation that maximizes the inside density of the clusters. The termination of the algorithm can be tuned heuristically in order to obtain different levels of cluster granularity, for instance at note level or according to different sources or audio classes [68].

Lastly, the computational complexity of the segmentation algorithm is $O(N^2)$, where N is the total number of frames. This depends on the fact that similarity and ranking values need to be computed for each couple of frames; in the case of long music recordings and lim-

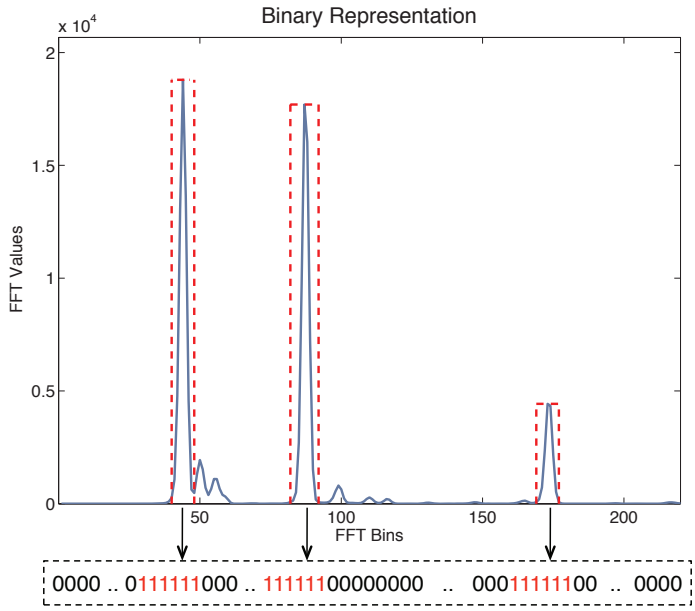


Figure B.3. Example of an audio frame converted into a binary string.

ited computational power, this may become a relevant issue. However, this problem can be overcome by segmenting short overlapping excerpts of a complete recording and then concatenating together the segments. A set of preliminary experiments on a collection of monophonic and polyphonic recordings of orchestral music showed that the segmentation process is robust to the size of the excerpts. That is, segmenting 5 seconds of music gives the same (sub)set of clusters as segmenting a longer excerpt of 10 or more seconds. Therefore, segmentation can be carried out efficiently by choosing a suitable length for the music excerpts.

B.2 From Bins to Bits

Given the segments obtained at the previous step, we propose to represent their frequency content in binary form, that is every bin is represented by a single bit[1]. Relevant features are highlighted by selecting the positions of the local maxima in the spectral representation. The general assumption is that similar performances are expected to have similar local maxima in the frequency, that is the dominant pitches should be in close positions. In principle it could be expected that similar songs may have similar spectra (i.e. in terms of energy absolute value); however, this assumption does not hold, because of the differences in performing styles, room acoustics, recording equipment, audio post processing, etc. that may occur.

For each frame, the procedure works by highlighting the bins corresponding to local maxima (i.e. intended as top of triangular shapes) in decreasing order, i.e. $\{p_1, p_2, \ldots, p_T\}$, where the highest peak corresponds to bin p_1. The number of selected maxima is computed

[1] We refer to the segments output by the segmentation process; however, the same bit-based representation can be applied to the original audio frames as well.

iteratively, by requiring that the sum of the energy components of the picked intervals is above a given threshold. The threshold is set as a fraction of the overall energy of the signal; empirically, we found that a percentage of 90% is a good trade-off between the number of excluded relevant peaks (false negatives) and included non relevant peaks (false positives). In particular, for a frame X with spectrum $\hat{X}$, the number of peaks is computed iteratively where the following condition must hold:

$$M = \text{argmax}_T \sum_{t=1}^{T} \hat{X}(p_t) \quad \text{such that} \quad \sum_{t=1}^{M} \hat{X}(p_t) \leq 0.9 \cdot \sum_{j=1}^{L/2} \hat{X}(j) \ , \tag{B.3}$$

where L is the length of the audio frame. Once M is computed all the frequency bins that correspond to peak positions $\{p_1, \ldots, p_M\}$ are set to 1, and all the others are set to 0.

In order to compensate windowing effects, the actual computation is slightly more complex, because each peak is associated to a frequency interval with the size of a quarter tone. Therefore, in the previous summations the contribution of surrounding frequency bins should be added to the term $\hat{X(p_t)}$; nevertheless, we prefer to show the simplified version for the sake of clarity. In addition, Figure B.3 exemplifies the approach: the solid lines depict the FFT of a frame, while the dashed rectangles highlight the selected intervals whose bins are mapped as 1s.

Similarity between binary representations of frames or segments can be done using logical operations [45]. Given two frames (segments), the complement of a bitwise XOR between all the corresponding bits gives a value equal to 1 for all bins that are likely to contain a spectral peak in both signals. Moreover, frame similarity can be computed as the number of 1s of a bitwise AND divided by the number of 1s of a bitwise OR, which can be considered as a simplified version of the cosine distance introduced in Section B.1. The fact that similarity can be computed only through counts and logical operations is expected to reduce the computational costs, although in specialized DSP architectures the cost of multiplications may not be dramatically different from simpler operations. Additionally, this representation can be used as the basis for a statistical modeling of the signal for a music identification task, as described in Section 2.2.6.1 (b); in particular, the binary descriptors could characterize the emission probabilities of the states in a hidden Markov model that represents the audio signal as a double embedded stochastic process through the time [68].

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] S. Agarwal. Ranking on graph data. In *Proc. of ICML*, pages 25–32, 2006.

[3] J.J. Aucouturier, F. Pachet, P. Roy, and A. Beurivè. Signal + context = better classification. In *Proc. of ISMIR*, pages 425–430, 2007.

[4] R. Baeza-Yates and B. Ribeiro-Neto, editors. *Modern Information Retrieval*. ACM Press, New York, NY, 1999.

[5] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Proc. of ICCV*, volume 2, pages 408–415, 2001.

[6] L. Barrington, R. Oda, and G. Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *Proc. of ISMIR*, pages 357–362, 2009.

[7] M.A. Bartsch and G.H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, 2005.

[8] J. Bello. Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats. In *Proc. of ISMIR*, pages 239–244, 2007.

[9] 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.

[10] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: a model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[11] T. Bertin-Mahieux, R.J. Weiss, and D.P.W. Ellis. Clustering beat-chroma patterns in a large music database. In *Proc. of ISMIR*, pages 111–116, 2010.

[12] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] D.M. Blei and M.I. Jordan. Modeling annotated data. In *Proc. of ACM SIGIR*, pages 127–134, 2003.

[14] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proc. of ACM MULTIMEDIA*, pages 391–400, 2010.

[15] E. Di Buccio, N. Montecchio, and N. Orio. A scalable cover identification engine. In *Proc. of ACM MULTIMEDIA*, pages 1143–1146, 2010.

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

[17] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Proc. of ECCV*, pages 350–362, 2004.

[18] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

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

[20] O. Celma. Foafing the music: Bridging the semantic gap in music recommendation. In *ISWC*, pages 927–934, 2006.

[21] O. Celma. *Music Recommendation and Discovery in the Long Tail*. PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[22] C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[23] Z.S. Chen and J.S. Jang. On the use of anti-word models for audio music annotation and retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, 8(17):1547–1556, 2009.

[24] E. Chew. *Towards a mathematical model of tonality*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, USA, 2000.

[25] F.Y.Y. Choi. Advances in domain independent linear text segmentation. In *Proc. of NAACL*, pages 26–33, 2000.

[26] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[27] E. Coviello, L. Barrington, A.B. Chan, and G. Lanckriet. Automatic music tagging with time series models. In *Proc. of ISMIR*, pages 81–86, 2010.

[28] T. Crawford, M. Mauch, and C. Rhodes. Recognizing classical works in historical recordings. In *Proc. of ISMIR*, pages 495–500, 2010.

[29] R.B. Dannenberg, W.P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo. The MUSART testbed for query-by-humming evaluation. In *Proc. of ISMIR*, pages 34–48, 2003.

[30] R.B. Dannenberg and N. Hu. Understanding search performance in query-by-humming systems. In *Proc. of ISMIR*, pages 232–236, 2004.

[31] 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 B*, 39(1):1–38, 1977.

[32] J.S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[33] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. of ECCV*, pages 349–354, 2002.

[34] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Proc. of NIPS*, pages 385–392, 2007.

[35] D.P.W. Ellis and G.E. Poliner. Identifying "cover songs" with chroma features and dynamic programming beat tracking. In *Proc. of IEEE ICASSP*, pages IV–1429–IV–1432, 2007.

[36] S.L. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *Proc. of IEEE CVPR*, pages 1002—1009, 2004.

[37] A. Flexer, D. Schnitzer, M. Gasser, and T. Pohle. Combining features reduces hubness in audio similarity. In *Proc. of ISMIR*, pages 171–176, 2010.

[38] G.D. Forney. The Viterbi algorithm. *Proceedings of IEEE*, 61(3):268–278, 1973.

[39] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proc. of ICMC*, pages 464–467, 1999.

[40] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. of VLDB*, pages 518–529, 1999.

[41] E. Gomez. *Tonal Description of Music Audio Signals*. PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[42] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech and Language processing*, 14(5):1783–1794, 2006.

[43] M. Goto and K. Hirata. Recent studies on music information processing. *Acoustical Science and Technology*, 25(4):419–425, 2004.

[44] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 5(14):1832–1844, 2006.

[45] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proc. of ISMIR*, pages 107–115, 2002.

[46] C.H. Harte, M. Sandler, and M. Gasser. Detecting harmonic changes in musical audio. In *Proc. of ACM MULTIMEDIA*, pages 21–26, 2006.

[47] M. Hoffman, D. Blei, and P. Cook. Content-based musical similarity computation using the hierarchical dirichlet process. In *Proc. of ISMIR*, pages 349–354, 2008.

[48] M. Hoffman, D. Blei, and P. Cook. Easy as CBA: A simple probabilistic model for tagging music. In *Proc. of ISMIR*, pages 369–374, 2009.

[49] M. Khadkevich, , and M. Omologo. Use of hidden markov models and factored language models for automatic chord recognition. In *Proc. of ISMIR*, pages 561–566, 2009.

[50] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner, and G. Widmer. Augmenting text-based music retrieval with audio similarity. In *Proc. of ISMIR*, pages 579–584, 2009.

[51] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *Proc. of ACM SIGIR*, pages 23–27, 2007.

[52] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 12(2):79–86, 1951.

[53] F. Kurth and M. Müller. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, 2008.

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

[55] P. Lamere and O. Celma. Music recommendation tutorial notes, 2008. ISMIR Tutorial.

[56] O. Lartillot and P. Toiviainen. A MATLAB toolbox for musical feature extraction from audio. In *Proc. of DAFx*, pages 1–8, 2007.

[57] E. Law and L. Von Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *Proc. of ACM SIGCHI*, pages 1197–1206, 2009.

[58] J. Li and J.Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075–1088, 2003.

[59] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. of ISMIR*, 2000.

[60] M. Mandel and D.P.W. Ellis. Song-level features and support vector machines for music classification. In *Proc. of ISMIR*, pages 594–599, 2005.

[61] M. Mandel and D.P.W. Ellis. Multiple-instance learning for music information retrieval. In *Proc. of ISMIR*, pages 577–582, 2008.

[62] C.D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[63] B. McFee and G. Lanckriet. Heterogenous embedding for subjective artist similarity. In *Proc. of ISMIR*, pages 513–518, 2009.

[64] T. Minka. Estimating a Dirichlet distribution. 2009. `http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/`.

[65] R. Miotto, L. Barrington, and G. Lanckriet. Improving auto-tagging by modeling semantic co-occurrences. In *Proc. of ISMIR*, pages 297–302, 2010.

[66] R. Miotto, N. Montecchio, and N. Orio. Statistical music modeling aimed at identification and alignment. In Z.W. Ras and A.A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, pages 187–212. Springer, Berlin Heidelberg, DE, 2010.

[67] R. Miotto and N. Orio. Automatic identification of music works through audio matching. In *Proc. of ECDL*, pages 124–135, 2007.

[68] R. Miotto and N. Orio. A methodology for the segmentation and identification of music works. In *Proc. of ISMIR*, pages 271–284, 2007.

[69] R. Miotto and N. Orio. A music identification system based on chroma indexing and statistical modeling. In *Proc. of ISMIR*, pages 301–306, 2008.

[70] R. Miotto and N. Orio. A probabilistic approach to merge context and content information for music retrieval. In *Proc. of ISMIR*, pages 15–20, 2010.

[71] N. Montecchio and N. Orio. A discrete filter bank approach to audio to score matching for polyphonic music. In *Proc. of ISMIR*, pages 495–500, 2009.

[72] S.R. Ness, A. Theocharis, G. Tzanetakis, and L.G. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proc. of ACM MULTIMEDIA*, pages 705–708, 2009.

[73] N. Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieva*, 1(1):1–90, 2006.

[74] E. Pampalk. *Computational models of music similarity and their application to music information retrieval*. PhD Thesis, Vienna University of Technology, Austria, 2006.

[75] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classificaton. In *Proc. of ISMIR*, pages 628–633, 2005.

[76] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc. of ACM MULTIMEDIA*, pages 570–579, 2002.

[77] G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proc. of ISMIR*, pages 115–120, 2006.

[78] J.C. Platt. Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[79] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[80] L.R. Rabiner. A tutorial on hidden Markov models and selected application. *Proceedings of IEEE*, 77(2):257–286, 1989.

[81] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

[82] N. Rasiwasia and N. Vasconcelos. Bridging the semantic gap: Query by semantic example. *IEEE Transactions on Multimedia*, 9(5):923–938, 2007.

[83] N. Rasiwasia and N. Vasconcelos. Holistic context modeling using semantic co-occurences. In *Proc. of IEEE CVPR*, pages 1889–1895, 2009.

[84] J.C. Reynar. *Topic Segmentations: Algorithms and Applications*. PhD Thesis, Computer and Information Science, University of Pennsylvania, USA, 1998.

[85] J. Serra, E. Gomez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.

[86] K. Seyerlehner, G. Widmer, and P. Knees. Frame level audio similarity – a codebook approach. In *Proc. of DAFx*, pages 349–356.

[87] J. Shifrin, B. Pardo, C. Meek, and W. Birmingham. HMM-based musical query retrieval. In *Proc. of ACM/IEEE JCDL*, pages 295–300, 2002.

[88] M. Slaney. Semantic audio retrieval. In *Proc. of IEEE ICASSP*, pages IV–1408–IV–1411, 2002.

[89] M. Slaney and M. Casey. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *IEEE Signal Processing Magazine*, 25(2):128–131, 2008.

[90] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *Proc. of ISMIR*, pages 313–318, 2008.

[91] M. Sordo, C. Laurier, and O. Celma. Annotating music collections: How content-based similarity helps to propagate labels. In *Proc. of ISMIR*, pages 531–534, 2007.

[92] S.S. Stevens, J. Volkman, and E.B. Newman. A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937.

[93] D. Tingle, Y. Kim, and D. Turnbull. Exploring automatic music annotation with "acoustically-objective" tags. In *Proc. of ACM ICMR*, pages 55–61, 2010.

[94] B. Tomasik, J.H. Kim, M. Ladlow, M. Augat, D. Tingle, R. Wicentowski, and D. Turnbull. Using regression to combine data sources for semantic music discovery. In *Proc. of ISMIR*, pages 405–410, 2009.

[95] B. Tomasik, P. Thiha, and D. Turnbull. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In *Proc. of IEEE ICASSP*, pages 437–440, 2010.

[96] C.F. Tsai and C. Hung. Automatically annotating images with keywords: a review of image annotation systems. *Recent Patents on Computer Science*, 1:55–68, 2008.

[97] D. Turnbull. *Design and Development of a Semantic Music Discovery Engine*. PhD Thesis, University of California, San Diego, USA, 2008.

[98] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. In *Proc. of ISMIR*, pages 225–230, 2008.

[99] D. Turnbull, L. Barrington, G. Lanckriet, and M. Yazdani. Combining audio content and social context for semantic music discovery. In *Proc. of ACM SIGIR*, pages 387–394, 2009.

[100] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. In *Proc. of ACM SIGIR*, pages 439–446, 2007.

[101] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on audio, speech, and language processing*, 16(2):467–476, 2008.

[102] R. Typke. *Music Retrieval based on Melodic Similarity*. PhD Thesis, Utrecht University, The Netherlands, 2007.

[103] G. Tzanetakis and P. Cook. MARSYAS: a framework for audio analysis. *Open Sound*, 30(4):169–175, 2000.

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

[105] N. Vasconcelos. From pixels to semantic spaces: Advances in content-based image retrieval. *IEEE Computer*, 40(7):20–26, 2007.

[106] D. Wang, T. Li, and M. Ogihara. Are tags better than audio features? The effects of joint use of tags and audio content features for artistic style clustering. In *Proc. of ISMIR*, pages 57–62, 2010.

[107] B. Whitman and D.P.W. Ellis. Automatic record reviews. In *Proc. of ISMIR*, pages 86–93, 2004.

[108] B. Whitman and R. Rifkin. Musical query-by-description as a multi-class learning problem. In *Proc. of IEEE MMSP*, pages 153–156, 2002.

[109] F. Xia, T.Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proc. of ICML*, pages 1192–1199, 2008.

[110] Y.H. Yang, Y.C. Lin, A. Lee, and H. Chen. Improving musical concept detection by ordinal regression and context fusion. In *Proc. of ISMIR*, pages 147–152, 2009.