UNIVERSITAT POMPEU FABRA

Semantic Annotation of Music Collections: A Computational Approach

Mohamed Sordo

TESI DOCTORAL UPF / 2011

Directors de la tesi:

Dr. Xavier Serra i Casals
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dr. Òscar Celma i Herrada
Gracenote, Emeryville, CA, USA

Dissertation submitted to the Department of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA,

with the mention of European Doctor.

Music Technology Group (http://mtg.upf.edu), Dept. of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

A Radia, Idris y Randa. Me siento muy orgulloso
de ser vuestro hijo y tu hermano.
A toda mi familia.

Acknowledgements

During these last few years, I had the luck to work with an amazing group of people at the Music Technology Group. First and foremost, I would specially like to thank 3 people regarding this dissertation. Xavier Serra, for giving me the opportunity to join the Music Technology Group, and for his wise advices in key moments of the thesis work. Òscar Celma, for being the perfect co-supervisor a post–graduate student can have. Whether it was for guidance or for publishing, he was always there. Fabien Gouyon, who would have been without any doubt the third supervisor of this thesis. I especially thank him for giving me the opportunity to join his research group in the wonderful city of Porto, as a research stay.

I also want to thank many other MTG people who have given me knowledge and moral support throughout these years. In no specific order, thanks to Perfecto Herrera, Hendrik Purwins, Rafael Ramírez, Dmitry Bogdanov and Joan Serrà for solving my machine learning doubts. To Pedro Cano, from whom the idea of automatic labeling using audio similarity was first taken. To Cyril Laurier and Martín Blech, for the work and fun in our joint publications and in the European Project PHAROS. To Nicolas Wack, for his amazing work providing tools and knowledge for efficiently extracting, transforming and classifying audio excerpts. To Oscar Mayor, for the long talks about the last trends in technology. To José Zapata and Jordi Sesmero, for being very good friends. To Justin Salamon, Luís Sarmento, Graham Coleman, Ferdinand Fuhrmann, Martin Haro, Piotr Holonowicz, Amaury Hazan, Ricard Marxer, Owen Meyers, Elena Martínez, Jordi Funollet, Jens Grivolla, Koppi, Eduard Aylon, Enric Guaus, Emilia Gómez, Andreas Beisler, José Pedro and Pablo García. Sorry if I forgot anyone. To the amazing MTG administration staff, Cristina Garrido and Alba Rosado. To Lydia García, who helped me to put all the paperwork in order. To my new MTG roommates, Sankalp Gulati, Gopala Koduri, and Sertan Şenturk. Sertan should know that I owe him a big one. To Robin Motheral, for proofreading some parts of this thesis. Robin is awesome.

This research was conducted at the Music Technology Group of Universitat Pompeu Fabra (UPF) in Barcelona, from September 2007 to March 2010 and from July 2010 to December 2011, and at the INstituto de Engenharia de Sistemas e Computadores (INESC) do Porto in Porto, Portugal, from April 2010 to June 2010. This work has been supported by an R+D+I scholarship from UPF, by the European Commission projects PHAROS (IST-2006-045035) and VARIAZIONI (ECP-2005-CULT-038264). The research stay at the INESC was funded by Ministerio de Educación in Spain as a mobility program of PhD students to obtain the mention of "European doctor".

Abstract

Music consumption has changed drastically in the last few years. With the arrival of digital music, the cost of production has substantially dropped. The expansion of the World Wide Web has helped to promote the exploration of many more music content. Online stores, such as iTunes or Amazon, own music collections in the order of millions of songs. Accessing these large collections in an effective manner is still a big challenge.

In this dissertation we focus on the problem of annotating music collections with semantic words, also called tags. The foundations of all the methods used in this dissertation are based on techniques from the fields of information retrieval, machine learning, and signal processing. We propose an automatic music annotation algorithm that uses content-based audio similarity to propagate tags among songs. The algorithm is evaluated extensively using multiple music collections of varying size and quality of the data, including a large music collection of more than a half million songs, annotated with social tags derived from a music community. We assess the quality of our proposed algorithm by comparing it with several state of the art approaches. We also discuss the importance of using evaluation measures that cover different dimensions; per–song and per–tag evaluation. Our proposal achieves state of the art results, and has ranked high in the MIREX 2011 evaluation campaign. The obtained results also show some limitations of automatic tagging, related to data inconsistencies, correlation of concepts and the difficulty to capture some personal tags with content information. This is more evident in music communites, where users can annotate songs with any free text word. In order to tackle these issues, we present an in-depth study of the nature of music folksonomies. We concretely study whether tag annotations made by a large community (i.e. a folksonomy) correspond with a more controlled, structured vocabulary by experts in the music and the psychology fields. Results reveal that some tags are clearly defined and understood both by the experts and the wisdom of crowds, while it is difficult to achieve a common consensus on the meaning of other tags. Finally, we extend our previous work to a wide range of semantic concepts. We present a novel way to uncover facets implicit in social tagging, and classify the tags with respect to these semantic facets. The latter findings can help to understand the nature of social tags, and thus be beneficial for further improvement of semantic tagging of music.

Our findings have significant implications for music information retrieval systems that assist users to explore large music collections, digging for content they might like.

Resumen

El consumo de la música ha cambiado drásticamente en los últimos años. Con la llegada de la música digital, el coste de producción se ha reducido considerablemente. La expansión de la Web ha ayudado a promover la exploración de mucho más contenido musical. Algunas tiendas musicales on-line, como iTunes o Amazon, poseen millones de canciones en sus colecciones. Sin embargo, acceder a estas colecciones de una manera eficiente es todavía un gran reto.
En esta tesis nos centramos en el problema de anotar colecciones musicales con palabras semánticas, también conocidas como tags. Los métodos utilizados en esta tesis están cimentados sobre los campos de recuperación de la información, la inteligencia artifical, y el procesamiento del señal. Proponemos un algoritmo para anotar música automáticamente, usando similitud de audio a nivel de contenido para propagar tags entre canciones. El algoritmo se evalúa extensamente usando múltiples colecciones musicales de distinto tamaño y calidad de los datos, incluyendo una colección de más de medio millón de canciones, anotadas con tags sociales derivados de una comunidad musical. Evaluamos la calidad de nuestro algoritmo mediante una comparación con algoritmos del estado del arte. Adicionalmente, discutimos la importancia de usar medidas de evaluación que cubren diferentes dimensiones; es decir, evaluaciones a nivel de canción y a nivel de tag. Nuestro algoritmo ha sido evaluado y se ha clasificado en altas posiciones en el concurso de evaluación internacional MIREX 2011. Los resultados obtenidos también demuestran algunas limitaciones de la anotación automática, relacionadas con las inconsistencias en los datos, la correlación de conceptos y la dificultad de capturar algunos tags personales con información del contenido. Esto es más evidente en las comunidades musicales, donde los usuarios pueden anotar canciones con cualquier palabra, sea esta contextual o no. Con el fin de abordar estas limitaciones, presentamos un amplio estudio sobre la naturaleza de las folksonomías musicales. Concretamente, estudiamos si las anotaciones hechas por una gran comunidad de usuarios concuerdan con un vocabulario más controlado y estructurado por parte de expertos en el campo. Los resultados revelan que algunos tags están claramente definidos y comprendidos tanto desde el punto de vista de los expertos como el de la sabiduría popular, mientras que hay otros tags sobre los cuales es difícil encontrar un consenso. Por último, extendemos nuestro previo trabajo a un amplio abanico de conceptos semánticos. Presentamos un método novedoso para descubrir conceptos semánticos implícitos en los tags sociales, y clasificar dichos tags con respecto a los conceptos semánticos. Los últimos hallazgos pueden ayudar a entender la naturaleza de los tags sociales, y por consiguiente ser beneficiales para una adicional mejora para la anotación automática de la música.

Resum

El consum de la música ha canviat dràsticament en els últims anys. Amb l'arribada de la música digital, el cost de producció s'ha reduït considerablement. L'expansió de la Web ha ajudat a promoure l'exploració de molt més contingut musical. Algunes botigues musicals on-line, com iTunes o Amazon, posseeixen milions de cançons a les seves col·leccions. No obstant, accedir a aquestes col·leccions d'una manera eficient és encara un gran repte.
En aquesta tesis ens centrem en el problema d'anotar col·leccions musicals amb paraules semàntiques, també conegudes com tags. Els mètodes utilitzats en aquesta tesi estan fonamentats sobre els camps de recuperació de la informació, l'intel·ligència artificial, i el processament del senyal. Proposem un algorisme per anotar música automàticament, utilitzant similitud d'audio a nivell de contingut per propagar tags entre cançons. L'algorisme s'avalua extensament utilitzant múltiples col·leccions musicals de diferent mida i qualitat de les dades, incloent una col·lecció de més de mig milió de cançons, anotades amb tags socials derivats d'una comunitat musical. Avaluem la qualitat del nostre algorisme mitjançant una comparació amb algorismes de l'estat de l'art. Addicionalment, discutim la importància d'utilitzar mesures de avaluació que cobreixen diferents dimensions, és a dir, avaluacions a nivell de cançó i a nivell de tag. El nostre algorisme ha estat avaluat i s'ha classificat en altes posicions en el concurs d'avaluació internacional MIREX 2011. Els resultats obtinguts també demostren algunes limitacions de l'anotació automàtica, relacionades amb les inconsistències en les dades, la correlació de conceptes i la dificultat de capturar alguns tags personals amb informació del contingut. Això és més evident en les comunitats musicals, on els usuaris poden anotar cançons amb qualsevol paraula, sigui aquesta contextual o no. Per tal d'abordar aquestes limitacions, presentem un ampli estudi sobre la naturalesa de les folksonomies musicals. Concretament, estudiem si les anotacions fetes per una gran comunitat d'usuaris coincideixen amb un vocabulari més controlat i estructurat per part d'experts en el camp. Els resultats revelen que alguns tags estan clarament definits i compresos tant des del punt de vista dels experts com el de la saviesa popular, mentre que n'hi ha d'altres sobre els quals és difícil trobar un consens. Finalment, estenem el nostre previ treball a un ampli ventall de conceptes semàntics. Presentem un nou mètode per a descobrir conceptes semàntics implícits en els tags socials, i classificar aquests tags pel que fa als conceptes semàntics. Les darreres troballes poden ajudar a entendre la naturalesa dels tags socials, i per tant ser beneficials per a una addicional millora de la anotació automàtica de la música.

Contents

List of figures

List of tables

CHAPTER 1

Introduction

We are entering an era of unprecedented choice. And that's a good thing.
—Chris Anderson (2006)

1.1. Motivation

Music is unquestionably a fundamental part of the society. It generates big communities of listeners and also artists. The expansion of the World Wide Web, together with the popularization of music production equipments, have helped to promote and to make known many artists and bands. Indeed, it enables any local garage band to control the creative process from the beginning till the end. On the other hand, music consumption has changed drastically in the last few years. Users are provided with large storage computers, media players, or more recently the ability to use the cloud for storing and accessing large music collections. Commercial systems — such as iTunes, Amazon, or Spotify — own music collections in the order of millions of songs, and are continuously growing.

Paradoxically, while this "paradigm" of infinite choices seems beneficial for the creation of more distinctive and complex music profiles, accessing such large musical collections — in terms of retrieval, browsing and recommendation — becomes more and more difficult.

One way to ease the access of large music collections is to keep annotations of all the music resources. Manual annotation of multimedia data, however, is an arduous task, and very time consuming. Automatic annotation methods, normally fine–tuned to reduced domains, such as music genre classification, are not mature enough to label with great detail any possible sound. Yet, in the music domain the annotation becomes more complex due to the time domain aspect. The purpose of making music easily accessible implies a condition of describing music in such a way that machine learning can uncover it (Pachet, 2005). Specifically, these two steps must be followed: to build music descriptions which can be easily maintained, and to exploit these descriptions to build efficient systems to access music and that help users finding music in large collections. There are a lot of ways to describe music content, but we can basically classify the descriptors in three groups: editorial meta-data, cultural meta-data, and acoustic meta-data (Pachet, 2005). As a paradigmatic example, the Music Genome Project is a big effort to "capture the essence of music at the fundamental level" by using over 400 attributes to describe songs. To achieve this, more than 40 musicologists have been annotating thousands of files since 2000. Based on this knowledge, a well–known system named Pandora[1] creates playlists by exploiting these human–based annotations. It is clear that helping these musicologists can reduce both time and cost of the annotation task.

Music autotagging is a very recent research problem. It is in fact encompassed within the research field of Music Information Retrieval (MIR), which, although having reached a certain level of maturity (Downie et al., 2009), is still in its early stages, when compared to research areas such as signal processing or speech processing. Indeed, the first international conference fully devoted to MIR, the International Society for Music Information Retrieval (ISMIR), was held in Plymouth, Massachusetts, in the year 2000. The first research works on automatic genre classification[2] date back to 2001 (Tzanetakis et al., 2001), and mood classification in 2003 (Liu et al., 2003). As for the music automatic tag classification, or music autotagging, the first research findings date back to late 2005 (Mandel & Ellis, 2005; Turnbull et al., 2006). Since then, several algorithms have been proposed for the task of automatic tagging from audio (Aucouturier et al., 2007b; Bertin-Mahieux et al., 2008; Burred & Peeters, 2009; Eck et al., 2008; Hoffman et al., 2009; Panagakis et al., 2010; Sordo et al., 2007; Turnbull et al., 2008b). Most of these approaches rely on the bag of acoustic features extracted from the audio[3], combined with Machine Learning algorithms — which classify or discriminate between the different concepts of music — and a set of previously labeled datasets, usually referred to as Ground Truth, or Gold Standard.

1.2. The problem of music autotagging

Music classification is a very complex and arduous task, specially due to the time domain aspect of the audio. Research questions about automatically classifying music in specific aspects, such as musical genre, instruments, or moods, are far from being completely solved. Music autotagging might be even harder than other music related classification task, in the sense that it includes all of the mentioned music aspects, whilst adding many other concepts — includ-

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

[2] As indexed by Google Scholar on December 11th, 2011, http://scholar.google.com/.

[3] based on the bag of words representation in classical Information Retrieval from text.

ing performance, cultural, geographic locations, decades, content interaction, etc. — that current acoustic features cannot capture with a high accuracy, or cannot capture at all. Moreover, many of these aspects tend to be correlated. For example, *synthesizers* or *beat machines* are usually correlated with *hip–hop* music, which is sometimes also characterized by using *strong male vocals* that produce *aggressive* music.

Current research on music autotagging emphasizes the use of contextual information to improve the automatic annotations. Some authors (Marques et al., 2011; Miotto et al., 2010; Ness et al., 2009) exploit the aspect of tag correlation, proposing two–stage algorithms which (1) learn the probabilistic/weighted relation between tags and audio, and (2) use the previous probabilities to generate improved annotations. Other authors (Barrington et al., 2009; Coviello et al., 2011; Knees et al., 2009; Mandel et al., 2011) focus on combining audio based annotations with contextual information (such as social tags, mp3 web logs, record reviews, etc.) to further improve the algorithm predictions.

Datasets

One of the crucial steps towards having a valid algorithm is the use of ground truth datasets. It is strongly encouraged that these datasets be:

- As balanced as possible. That is, there should be a similar number of instances per each tag.
- As complete as possible. They should describe music excerpts in all possible ways.
- Shared among the scientific community, in order to encourage exhaustive evaluation and comparison among the different algorithms.

The first proposed Ground Truth datasets relied on controlled vocabularies as defined by a group of music experts. These vocabularies are usually structured in hierarchies or taxonomies, and include well–known concepts of music such as musical genres, moods or instrumentation. However, there exists no consensual taxonomy for music (Pachet & Cazaly, 2000).

In the last few years, there has been a growing interest in using social networks for sharing individual tastes. Social music websites, such as Last.fm[4], allow users to inform what they are currently listening to in their music devices. Interestingly, it also allows users to tag music items (artists, albums and tracks) either for personal organizational purposes, or to communicate their musical taste. The combination of the annotations provided by thousands of music users leads to the emergence of a large body of domain specific knowledge, usually referred to as folksonomy. By the time of submitting this dissertation,

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

Last.fm has collected over 11 million artist annotations, and 33 million track annotations, built from over 2.9 million distinct tags[5]. This shared knowledge enables the creation of community–based Ground Truth datasets that are useful for the task of music autotagging[6].
One of the main advantages of using folksonomies is that they ideally encompass all possible "ways to talk about music", including both *expert* and *non–expert* points of view. However, the simplicity and user-friendliness of community-based tagging imposes a toll: there is usually no way to *explicitly* relate tags with the corresponding music facets. For instance, a user may use the tag *Bulgarian hip–hop* for songs such as "West Side Na Slonski Dozi" or artists like "Zelenata Kamora", while not explicitly expressing that this tag corresponds to a music genre (hip–hop) from a specific geographic location (Bulgaria). An autotagging algorithm, thus, might have no clue or prior knowledge about what kind of tags it is predicting or learning.

1.3. Our approach

This thesis addresses the problem of music tagging in different aspects. Figure 1.1 depicts the basic elements covered in the thesis. First, we propose an automatic approach to tag music from the raw audio, using content–based acoustic similarity as a way to propagate tags. Then, we tackle the problem of prior knowledge about semantic words, that is to say, we infer semantic facets and assign them to the tags, in order to improve the predictions made by our autotagging algorithm.

Automatic annotation from audio

We propose an automatic tagging algorithm that differs from state of the art approaches in two aspects. Firstly, it differs on the audio features. We use a wide range of audio features that includes timbre information (Mel Frequency Cepstral Coefficients, or MFCC) and spectral information, but also information related to tonality (Gomez, 2006; Serra et al., 2008), rhythm (Gouyon, 2005), energy (bark bands) and high level features (such as genres, moods, danceability, etc.). Secondly, we use the concept of audio similarity to propose tags. A weighted vote nearest neighbor classifier (or k-NN) assigns a tag to a new audio excerpt based on the distance measure between the audio excerpt and a previously labeled dataset (Ground Truth dataset). Figure 1.2 illustrates this process[7].

[5] This information was obtained through personal communication with Mark Levy, MIR team leader at Last.fm (2011).

[6] See for instance the recently published One Million Dataset (Bertin-Mahieux et al., 2011).

[7] Figure borrowed from the Music Recommendation tutorial, ISMIR 2007 (Celma & Lamere, 2007).

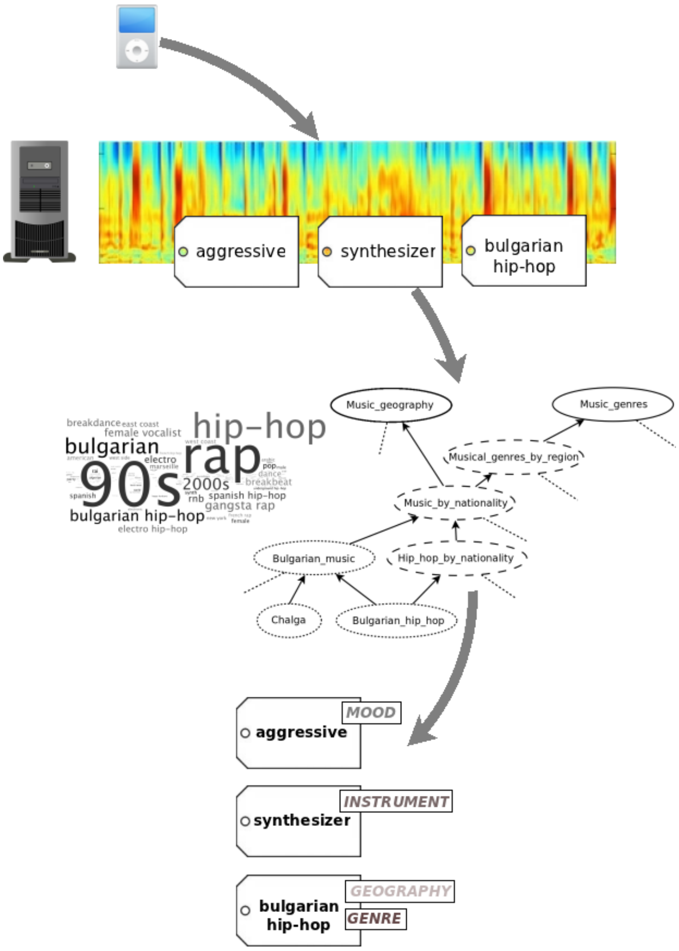

Figure 1.1: Diagram depicting the key parts of the thesis.

The distance measure is built upon the feature representation of each audio excerpt. Some of the advantages of using a memory–based classifier include:

1. There is no need for designing and learning or re–learning tags each time a tag or a song is added to the Ground Truth dataset.

2. Experimental results in this thesis show that the algorithm is scalable to music collections of a considerable size, comparable to current music collections in the music industry.

Additionally, in order to open a discussion regarding the performance of different classifiers, we propose a modification of our k-NN algorithm, which takes into account the class (i.e. tag) of the music tracks. Basically, it computes a centroid (Han & Karypis, 2000; Kim et al., 2006; Park et al., 2003) for each tag

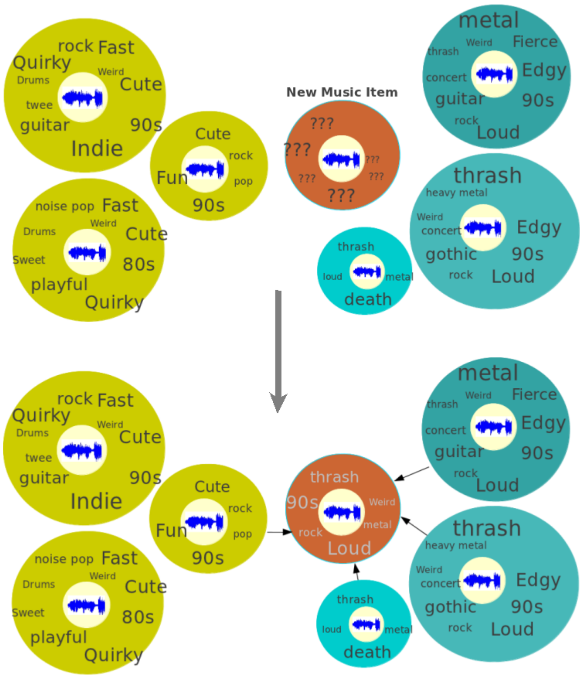

Figure 1.2: Figure that depicts the process of propagating tags to a new song from nearest neighbors in the "acoustical space".

in the ground truth dataset. The classification is then accomplished by calculating the minimum distances between the feature representation of the music excerpt and the tag centroids. Experimental results in Chapter 3 show that our algorithms perform as well as, or better than the state of the art approaches — which use more complex and time and resource consuming algorithms — in tasks such as music annotation and retrieval.

Folksonomies and Taxonomies

Music folksonomies have an extensive tag coverage, while being updated regularly; whereas taxonomies have a more precise and structured vocabulary, but rarely updated, as defined by music experts. In this sense, it seems that autotagging algorithms can only work with the latter option. One of the research questions that we try to answer in this thesis is the following: can a music autotagging algorithm rely on social tags as a Gold Standard? In other words, can we build data models from music folksonomies that can be useful for annotating music? Focusing on a large collection of tags crawled (i.e, extracted) from the Last.fm website, we first study two key aspects of music: musical genres and moods, since they are two important aspects when defining and classifying music. We show in exhaustive experiments that the knowledge inherent in the use of these social tags can be comparable to a controlled,

structured vocabulary defined by a group of music experts.
Based on the findings mentioned previously, we extend our research to a wide variety of music facets, including very specific and contextual information, such as *music festivals*, *record labels*, or *music competitions.* More concretely, we approach an essential question that is relevant to bridge the semantic gap between what humans perceive about music, and what pure content–based approaches to music classification use for discriminating between different classes (tags): is it possible to automatically infer the semantic facets inherent to a given music folksonomy? If so, is it then possible to classify tags from that music folksonomy with respect to the inferred semantic facets?
We propose an automatic method for (1) uncovering the set of semantic facets implicit to the tags of this music folksonomy, and (2) classify tags with respect to these facets. We anchor semantic facets on meta-data of the semi-structured repository of general knowledge Wikipedia. Our rationale is that as it is dynamically maintained by a large community, Wikipedia should contain *grounded* and *updated* information about relevant facets of music, in practice.

1.4. Outline of the Thesis

This thesis is structured as follows: Chapter 2 introduces the formalization and framework of music automatic tagging, referencing related work. Once the music autotagging has been introduced, Chapter 3 presents our proposed autotagging approach. It encloses an extensive evaluation against several Ground Truth datasets, from different sources, and compares our approach to a variety of state of the art autotagging methods. We exploit semantic aspects of music tags from folksonomies in Chapters 4 and 5, which will provide the possibility of categorizing tags and hence adding prior knowledge to the music autotagging algorithm. Finally, Chapter 6 discusses open issues and future work, as well as drawing some conclusions. To summarize the outline of this thesis, Figure 1.3 extends Figure 1.1 showing the corresponding chapters for each element of the thesis.

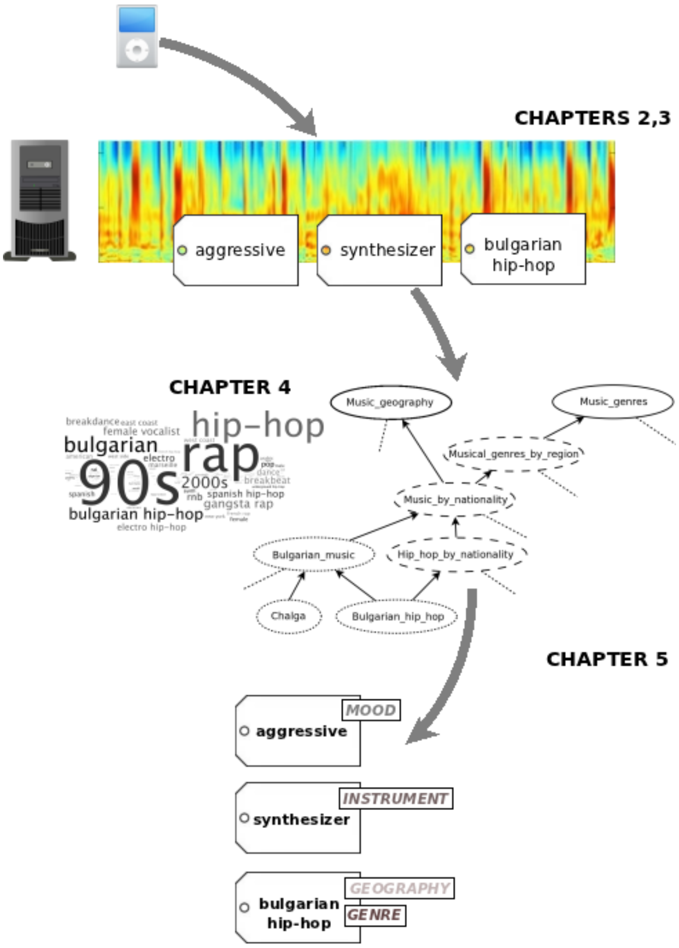

Figure 1.3: Extension of Figure 1.1 that adds information about the corresponding chapters.

CHAPTER 2

Framework of Automatic Tagging

2.1. Introduction

Nowadays, there is a vast amount of digital multimedia material available on the World Wide Web and on different types of digital storage media. For that reason, there is a need to organize and make this content easy to search, navigate, filter and retrieve in an efficient way. Searching in digital libraries has been studied for several years, mostly using text–based methods. These methods can be complemented with new strategies of retrieval, like those focused on content–based descriptors — extracted directly from the music files. However, these descriptors do not refer to any object in the real word, so that means that music is not strictly a type of knowledge. Another way of describing music, usually called meta–data, could help — in combination with the music content descriptors — to create some musical knowledge management, classification and representation.

The purpose of making all music easily accessible implies a condition of describing music in such a way that machine learning can uncover it, as (Pachet, 2005) states. Specifically, these two steps must be followed:

- Build descriptions of music easy to maintain, and
- Exploit these descriptions to build efficient music access systems that help the users finding music in large collections.

There are several ways for describing music content, but we can basically classify the descriptors in three groups (Celma, 2006; Pachet, 2005):

- Editorial meta–data: this kind of meta–data is obtained by the editor. Editorial meta–data includes songs and albums, but also information about artists. It can be either objective (song name, artist name, etc.) or subjective, like artists' biographies[1], genre information, etc. Depending on the nature of the human source, editorial meta–data could be also described as:

- prescriptive, where the information is decided by well–defined experts.
- non–prescriptive, where the information is classified based on collaborative scheme (a community of users).

- Cultural meta–data: the meta–data is obtained by the environment or culture. The information is not explicitly entered in an information system, rather is calculated using user profiles — also known as the so–called collaborative filtering. However, it does not depend only on these profiles — since it is very poor — but on other sources like search engines, encyclopedias, music radio programs, etc. The techniques — borrowed from natural language processing — are most of them based on co–occurrence analysis: associate items that are closer in some sense, for example, similar in genres, etc.

- Acoustic meta–data: obtained by the analysis of the audio file (no other kind of information is used), i.e, the content descriptors of the sounds. The intention is to have purely objective information about the music files. The descriptors can be either Tempo (in bps[2]) or other more complex descriptors like rhythm, timbre, instrument recognition (Herrera et al., 2005), etc.

We can see these three groups as different points of view of music annotation (meta–data). Moreover, recent research in Music Information Retrieval (MIR) is heading towards the use of perception and user contextual information. This kind of meta–data is obtained by exploring *various external factors that influence how a listener perceives music* (Schedl & Knees, 2011). These factors encompass many types of user contextual information, including environment (people, weather, noise, etc.), personal (physiological or psychological), task (activities such as sports or driving), social (friends, relatives, social networks, etc.) and spatio–temporal (location, time, etc.) (Göker & Myrhaug, 2002). Hence, if we take into account more than one point of view at a time, the result could be a better description of music.

2.2. Obtaining tags

We can annotate music files by means of tags, but what does this *word* mean? Tags are keywords, category names, or meta data that describe web content.

[1] Personal description of a human is almost always subjective information.

[2] beats per second.

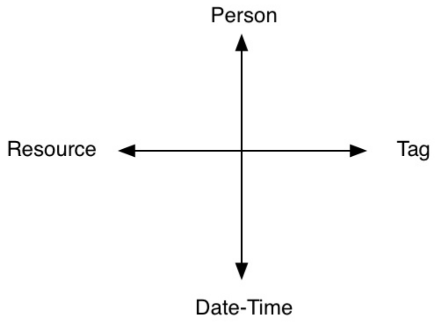

Figure 2.1: The tagging process.

Tags[3] can be whatever words that describe web content for users. But their job is not to organize all the information over the world wide web into tidy categories, rather it is to add value to the huge amount of data available nowadays (Beckett, 2006). Tagging is then a process to describe web content using these tags. This process is actually a combination of 4 entities, as shown in Figure 2.1.

- Person: who performs the operation, also called tagger.
- Tag: set of tags being used.
- Date–Time: when the tagging process was performed.
- Resource: the resource URI being described.

In the music field, tags are keywords that describe the audio files, and resource is the audio file itself.
Tags can be obtained in four different ways (Bertin-Mahieux et al., 2010; Turnbull et al., 2008a): conducting human surveys[4] (Turnbull et al., 2007a), deploying games with a purpose (Law et al., 2007; Mandel & Ellis, 2007; Turnbull et al., 2007b), collecting web documents (Schedl & Pohle, 2010; Whitman & Ellis, 2004) or harvesting social tags (Lamere, 2008; Levy & Sandler, 2008).

2.2.1. Web documents

The Web contains a huge amount of music–related content. This content, as other type of web content, is very heterogeneous in terms of data structure and information (Knees, 2010). Due to this heterogeneity, web data is also noisy, containing many irrelevant documents or irrelevant text contents inside

[3] In the rest of this thesis, both “tag” and “label” words will refer to the same concept.

[4] hiring musical experts to annotate music can be regarded as a special case of human surveys.

the documents (Levy & Sandler, 2007). The challenge lies on how to efficiently retrieve relevant data from the Web. Several contributions have been made to this task within the MIR community (Celma et al., 2006; Ellis et al., 2002; Knees, 2007; Knees et al., 2006, 2007a,a; Knees & Widmer, 2007; Schedl et al., 2005a,b, 2006; Whitman & Ellis, 2004; Whitman & Lawrence, 2002; Whitman & Rifkin, 2002).

For instance, Celma et al. (2006) exploited the semi-structured data from MP3 web logs, which contain links to audio content and some textual information referring to that content.

Whitman & Ellis, on the other hand, crawled websites for music artist–related information. In (Whitman & Lawrence, 2002) the authors proposed some methods for unsupervised learning of unstructured music profiles retrieved from the web, with the purpose of understanding the "semantic profile" of an artist, through a "*feature space that maximizes generality and descriptiveness*". These methods can help to infer artists' descriptions, represented as vector spaces, and similarity between artists by means of a peer–to–peer similarity.

(Whitman & Rifkin, 2002) presented a query–by–description (QBD) system that makes use of language processing, information retrieval and machine learning technique to answer queries such as "rock with guitar riffs". Their system treats the relation between web–based descriptions and music content as a 'severe multiclass' problem, using regularized least–squares classification (RLSC) (Rifkin et al., 2003). In a posterior work, Whitman et al. (2003) extended this technique by using a "linguistic expert", Wordnet[5], a lexical database, for finding parameter spaces that would help to describe better and more precisely the artists' descriptions.

Knees et al. (2008) queried search engines and extracted knowledge from the retrieved documents. The queries are of the form *artist music*, *artist album name music review* or *artist song music review*. The authors collected the top 100 hits and ranked them with a novel algorithm called *rank-based Relevance Scoring* (RS), which the authors have shown to outperform algorithms which use classical vector space representations (Knees et al., 2007b).

2.2.2. Surveys

Another way for obtaining tags for music is by conducting a survey, either for commercial or research purposes. As a paradigmatic example, the Music Genome Project is a big effort to "capture the essence of music at the fundamental level" by using over 400 attributes to describe songs. To achieve this, more than 40 musicologists have been annotating thousands of files since 2000. Based on this knowledge, a well–known system named Pandora[6] creates playlists by exploiting these human–based annotations. This approach, though, suffers from several drawbacks: On one hand, as Turnbull et al. (2008a)

[5] http://wordnet.princeton.edu/

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

points out, each song can take up to 330 minutes to annotated and ~ 15000 songs are annotated every month, which seems insufficient given the (growing) amount of digital music available nowadays. On the other hand, the company spends such an effort, in terms of time, money and human resources, to use the resulting annotations for commercial purpose, making it unfeasible for them to make these annotations public, not even for research purpose.
To overcome this issue, a group of researchers from the Computer Audition Lab in San Diego (Turnbull et al., 2007a, 2008b) have conducted a survey (paying undergraduate students to fill out the questionnaire) on a collection of 500 western popular songs (from 500 different artists, in an effort to maximize variation). Each tag annotation is confirmed/voted by three different listeners. The resulting dataset is given the name of CAL500.
As one can observe in both cases, commercial or research surveys suffer from a serious drawback: the amount of resources needed to perform such surveys make annotating large music collections an unfeasible task.

2.2.3. Games with a purpose

In the last few years, there has been a growing interest in collecting meta-data for multimedia resources using games (Von Ahn, 2006). Above all proposed systems, the most known game is ESP game, proposed by Von Ahn & Dabbish (2004). The ESP game is a double player game where each player is presented an image (the same as their opponent) and she is asked to start describing the image using words. When the two players agree in a description, they are rewarded with points. In the music domain, four games have been proposed:

TagATune (Law et al., 2007), a double–player game where each partner is presented with a tune (a song or a sound). The players start describing the tune from a specific category, defined a priori or set by one of the players. Based on this description, the players have to decide if they agree on whether they are annotating the same tune or not.

Major Miner (Mandel & Ellis, 2007) is a non–paired off line game, where a player requests a new music clip and annotates the clip with tags. The player wins points based on the originality and agreement of the tags.

ListenGame (Turnbull et al., 2007b), a multi-player on line game, where a player is presented with a music clip and a list of semantic words, from different categories, such as genre, mood, instrumentation, etc. The player is then asked to check the best and worst words describing the music piece. The player receives immediate feedback to see his agreement amongst other players. The authors propose also a freestyle mode of the game, where the players, given a specific category, are allowed to introduce semantic keywords that better describe the song.

MoodSwings (Kim et al., 2008), a double player game, where each player is given 5 short music clips, drawn from a music database of popular music, `uspop2002` (Ellis et al., 2003), and a game board representing the continuous 2–dimensional representation of moods (valence and arousal). The player is then asked to dynamically (every 1 second) express the mood of the music playing by moving the cursor of their mouse in the continuous space defined by the board. The player wins points when her cursor overlaps with her partner.

The motivation behind these kind of games for purpose, as compared to human surveys, is that while still rewarding the human annotators (in this case intrinsically), they overcome the problem of cost in terms of money. Two years after the presentation of their game, the authors of TagATune released a database named Magnatagatune[7], containing tags for more than 20000 songs. Nevertheless, these games suffer from some problems. First, a player can game the system, using words to intentionally mislead other players. This problem though can be easily tackled by keeping track of the user behavior. Second, some years after the release of these games, it is still not clear how they can scale up to hundreds of thousands or even millions of songs, an average dataset size in the music industry.

2.2.4. Social tags

As mentioned earlier, tags are keywords, category names or meta-data that describe web/multimedia content. These tags can be either obtained from a controlled vocabulary (Taxonomy) or by putting no restrictions on the vocabulary, that is, tags as free text. When these "free text" tags are introduced by users (usually non–experts) of any system to describe a content, they are known as social tags (Lamere, 2008). A usual representation of social tags is on the form of a tag cloud. A tag cloud is a visual representation of keywords in web or multimedia content. Different font sizes or colors are used for describing the importance of a keyword in the whole vocabulary. For instance, Figure 2.2 illustrates a tag cloud with the top most used tags in Last.fm.

The main shortcoming of social tags is sparseness. Few tags are applied to a large number of audio items, whilst most of the tags are very rarely used. Indeed, Lamere (2008) states that in Last.fm a track is annotated in average with 0.26 tags. However, as Lamere points out, even if tags from a non expert user may suffer from different problems, such as the cited sparseness, polysemy, Spam, noise, etc., when these tags are combined with other tags from thousands of other users a rich and complex view may emerge. This complex view is often referred to as Folksonomy.

[7] `http://tagatune.org/Magnatagatune.html`

Figure 2.2: Tag cloud of Last.fm top tags.

Motivations for tagging

An interesting point to take into account is that “why people started using tags?” or “which are the motivations that led users to tag digital objects, in our case, multimedia objects?”

To answer such questions, Ames and Naaman (Ames & Naaman, 2007) have developed an experiment and arrived to offering a taxonomy of motivations for annotation. Their work is focused on annotating images, but it can be useful for other kind of multimedia objects. The taxonomy consists of 2 dimensions, “sociality” and “function”, as it can be observed in Table 2.1. Sociality refers to the purpose of the tags, either for personal use (self) or for other users. The function dimension distinguishes the fact of using tags for organizing content or — as a new contribution from the authors — for communicating some additional context to the multimedia objects.

Table 2.1: A taxonomy of tagging motivations.

	Organization	**Communication**
Self	▪ Retrieval, Directory ▪ Search	▪ Context for self ▪ Memory
Social	▪ Contribution, attention	▪ Content descriptors ▪ Social signaling

In the same way, Lamere (2008) describes a list of motivations for tagging music, which shares, with more or less details, most of the motivations with the previous taxonomy, and adding a new motivation: play & competition. We now describe more in detail some of these motivations, for the specific field of music tagging.

Self–organization. People tag music content in order to ease the task of search and retrieval of this content for themselves. For example, a user might tag a song with the genre "Hip–Hop" or "To listen" so they can access this song in a future.

Social–organization. Willingness of making music easily findable by others, either by reinforcing tags which users think represent better the music piece, or by proposing new tags that contradict some of the previously assigned tags (Bertin-Mahieux et al., 2010; Lamere, 2008), in an effort to clean the tag cloud from undesirable or misleading tags. For instance, a user might tag an artist "not Rap", because she thinks the artist is not representative of such genre tag.

Social–communication. People tag music content in order to add some context for other people, or by signaling themselves in their social community, that is, showing their music taste to other people. For instance, a user might tag an artist "sxsw" or "seen live", to show some contextual information about the type of music that the user follows or enjoys at a concert. Another way of social communication is when a user expresses her opinion about a music resource, for example, using words such as "awful" or "amazing".

Table 2.2 summarizes the benefits and drawbacks of using each one of the different alternatives to obtaining tags for music. While crawling web documents allows to obtain tags in an automated fashion, it undergoes different problems, such as noise, popularity and tags at artist level instead of song level. Manual approaches, such as conducting surveys or deploying games, try to overcome the problems of the previous approach (getting a more consistent vocabulary, mixing known and unknown music and getting tags at song level (Turnbull et al., 2008a)), but still suffer from two big problems. First, they are not scalable to an average music database size found nowadays, in the order of thousands or millions of songs. And second, they are expensive in terms of cost (money, human resources and time, in the case of surveys), or the lack of motivation (popularity of the game). Harvesting social tags seems (able) to solve the problem of cost and motivation (a different motivation than winning a game), and based on recent research (Laurier et al., 2009b; Levy & Sandler, 2008; Sordo et al., 2008), when a folksonomy emerges, it tends to follow an inherent structure and in some cases it can be comparable to expert annotations. Collecting tags via social networks is also a scalable approach. By the time of writing this thesis, the social web Last.fm has collected over 11 million artist annotations, and 33 million track annotations, built from over 2.9 million distinct tags, artists and albums. Nevertheless, besides the problem of vocabulary (as in the case of web documents) and malicious tagging, the main

Table 2.2: A list of pros and cons of using different approaches to obtaining tags for music.

	Pros	**Cons**
Web Documents	▪ Automatic way of obtaining tags ▪ Linear/scalable	▪ Noise ▪ Tags at artist level instead of song level ▪ Not clear if the words are referring to the song/artist itself ▪ Polysemy / Synonymy ▪ Popularity bias
Human Surveys	▪ Large, but semi–structured vocabulary ▪ Consistent (agreement)	▪ Expensive in terms of money, human resources and time ▪ Non-attention ▪ Expert vs. non–expert ▪ Not scalable
Games	▪ Contribution, rewarding, attention ▪ Large, but semi–structured vocabulary	▪ Motivation/popularity of the game ▪ Gaming the system (malicious/misleading) ▪ Not scalable
Social Tags	▪ Easy and pleasurable (cheap) ▪ No restrictions on the vocabulary ▪ No hierarchy ▪ No limits on the number of tags per resource	▪ Cold start ▪ Polysemy / Synonymy ▪ Hacking / SPAM / Malevolous tagging

drawback is again the popularity bias. While popular artists and songs get hundreds of tags, new or unpopular music resources do not get any tags or few tags. The term popularity in this case has a different "effect". According to Lamere (2008), the typical tagger in social websites has a sense of popularity that may differ considerably from the music sales, a classic measure of popularity in music. In order to face all or some of the limitations of the previous approaches, researchers in the MIR community have proposed an automatic alternative for tagging music based on the acoustic description of the audio itself. This approach, often called autotagging (automatic tagging), extracts audio–related features from the audio and tries to infer tags for new music, either by learning the relation of tags with the audio to tag new, unseen music; or by using the features to create a similarity distance and proposing tags to new music pieces based on their neighborhood. This approach is described more in detail in the following section.

2.3. General framework

Different algorithms have been proposed for the task of music autotagging, though, in general, they follow the same structure depicted in Figure 2.3. The main structure is built upon previous MIR related tasks, such as genre (McKay & Fujinaga, 2004; Tzanetakis & Cook, 2002) or mood classification (Liu et al., 2003). In fact, many academicians see or treat autotagging as a generalization of the genre or mood classification to any type of words (tags).

An autotagging algorithm receives a Ground Truth composed of audio related features and tags as input, and produces models as output. These models are learned by using any of the state of the art Machine Learning algorithms (Alpaydin, 2004). The models will be then tested against a Test dataset[8] in order to check how well the algorithm performs. These models will be then used to propose tags to new, unseen music.

In the remainder of this subsection we focus more in detail on each one of the components which constitute the basic structure of an autotagging method, underlining related work in the state of the art.

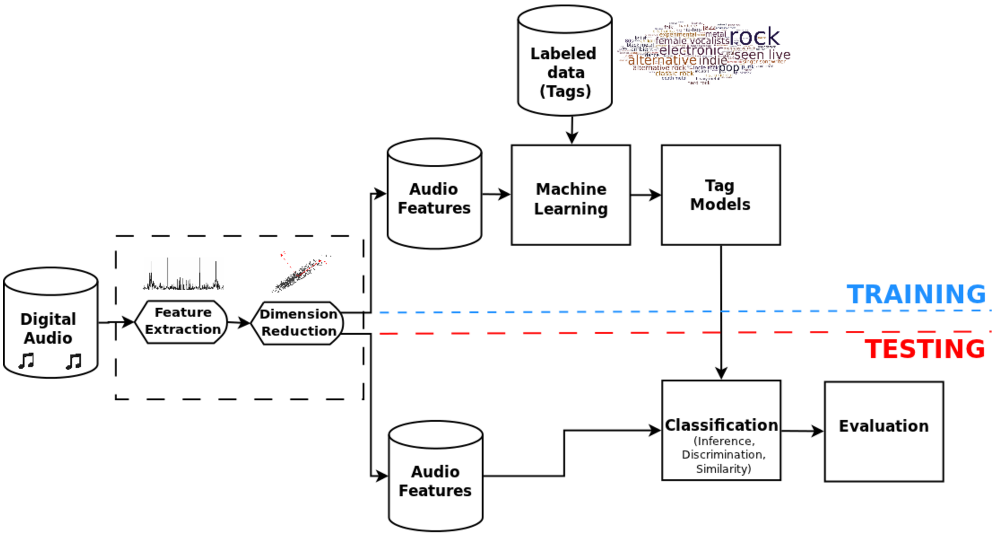

Figure 2.3: General framework for automatic tagging of audio.

[8] which can be extracted from the Ground Truth, that is, splitting the Ground Truth into a training an a testing set; or defined separately.

2.3.1. Feature extraction

Cook (2001) states that the musical aspects that humans use to describe music are pitch, loudness, duration and timbre, but also with terms such as style or texture. Similarly, Orio (2006) summarizes the most important facets of music as: timbre, orchestration, acoustics, rhythm, melody, harmony and structure.

Several approaches exist for representing acoustic description of audio as input for a machine learning algorithm. Most of them, though, rely on the "bag of frames" representation (Aucouturier et al., 2007a), taking advantage of the "bag of words" representation from the classic text Information Retrieval field (Baeza-Yates et al., 1999).

Audio features are normally extracted from a waveform representation of the digital audio files. The aim is to have a compact representation of audio that can help to represent the aforementioned key aspects of music. The audio features can range from low level (closer to the signal) to high level (close to human semantics) description of music. They are often captured on a short-time frame-by-frame basis, using half–overlapping windows (Gaussian, Hanning, Hamming) of short duration (typically 46ms–50ms). It is out of scope of this thesis the description of all the common audio features in the literature. A more detailed explanation on how most common audio features are computed can be found in (Gouyon et al., 2008). These features are then aggregated to a list or "bag" (hence the name of "bag of frames"). The bag of audio features is further compressed by means of random sub-sampling (Turnbull et al., 2008b) or summarizing, using statistics such as mean, variance and derivatives. The bag of frames approach has been widely used in the Music Information Retrieval domain (Aucouturier et al., 2007a). A large list of publications discuss the application of this approach to specific music classification problems, such as genre (Guaus, 2009; McKay & Fujinaga, 2004; Tzanetakis & Cook, 2002) mood (Laurier, 2011; Liu et al., 2003), artist (Mandel & Ellis, 2005) or tag classification (Bertin-Mahieux et al., 2008; Hoffman et al., 2009; Sordo et al., 2007; Turnbull et al., 2008b).

The most prominent audio features in MIR research are the Mel–Frequency Cepstral Coefficients[9], which have shown to be of a high importance for Speech Processing, since they tend to capture the human auditory system's response more closely (Rabiner & Juang, 1993). These features were then first used by Logan (2000) for music modeling.

In the case of automatic tagging, for instance, Mandel & Ellis (2005) used the 20 first MFCC coefficients to describe the audio, taking the recommendation from Aucouturier & Pachet (2004), who used this specification to improve timbre similarity, and more recently to propose an improved method of classification by combining signal and context (Aucouturier et al., 2007b).

More recently, Mandel & Ellis (2008) used additional audio features related to temporal information, following (Rauber et al., 2002). This temporal in-

[9] from now on, MFCC's.

formation, as the authors state, "*summarizes the beat, tempo, and rhythmic complexity of the music in four different frequency bands*". These bands are represented by a 200-dimensional feature vector, which are then combined with 180 unique elements from the covariance matrix of 18–MFCC's, resulting in a 380–dimensional feature vector.

Turnbull et al. (2008b) computed time series of MFCC–Delta feature vectors, by moving through a half–overlapping short time window (23ms) over the whole music file. They took the 13 first MFCC coefficients, and computed the first and second instantaneous derivatives, resulting in a 39–dimensional feature vector per frame, which means a total of 5200 feature vectors per minute of audio. Empirical results though made them randomly subsample the audio file and represent it with 10,000 feature vectors. The authors state that this process reduces time computation without suffering from any significant decrease in overall performance.

Similar to Turnbull et al. (2008b), Hoffman et al. (2009) used the 10,000 39–dimensional MFCC–delta feature vectors per song, as distributed with the CAL500 dataset (Turnbull et al., 2007a)[10]. Additionally, Hoffman et al. (2009) vector–quantized the MFCC–delta features instead of using them directly. The result of vector quantizing the deltas is a discrete "bag of codewords" representation, a considerably reduced feature set, but, as the authors state, it is still effective.

Bertin-Mahieux et al. (2008), on the other hand, used 20-MFCC coefficients, plus 176 autocorrelation coefficients and 85 spectrogram coefficients sampled by Constant–Q transform. These features are computed over windows of 100 ms size, with a 25% overlap. To reduce dimension, they aggregated features by taking means and standard deviations over 5–second windows.

2.3.2. Dimension reduction

The feature extraction process can provide the classifier a large amount of data. In most of the cases, not all of the computed descriptors provide useful information for classification.

There exist different techniques to reduce the dimension of feature vectors according to their discrimination power. Dimension reduction techniques can be classified in *Feature selection* (such as CFS Subset Evaluation (Hall, 1998), Entropy-based algorithms (Witten & Frank, 1999) or Gain ratio (Quinlan, 1986)), and *Feature transformation* (Principal Component Analysis, Linear Discriminat Analysis, Independent Component Analysis, Non-negative Matrix Factorization, Relevant Component Analysis). We briefly present a relevant dimension reduction algorithm for this thesis.

[10] the features in this dataset are provided randomly, which means no temporal information rather than the 69ms per each feature vector

Principal Component Analysis (PCA). Principal Compnent Analysis is an unsupervised statistical technique for identifying patterns in data (Jolliffe, 2002), specially for high–dimensional data, where graphical representation is unfeasible. In the case of music information retrieval, given a d–dimensional vector representation of each audio piece, PCA proceeds by projecting the original space (of the whole music dataset) to a new subspace, whose basis vectors represent the maximum–variance directions in the original space. Let X be a d–dimensional representation of the audio features (of the whole dataset), and let Φ represent the transformation that maps X onto an f–dimensional feature subspace (normally $f < d$). The new feature vectors $y_i (\in \mathbb{R}^f)$ are computed as:

$$y_i = \phi_i^T (X - \mu) \tag{2.1}$$

The mean is subtracted from X to center the data in the origin. Φ is a deterministic matrix whose columns are the Eigenvectors ϕ_i, which are obtained by solving the Eigendecomposition:

$$\Sigma_x \phi_i = \lambda_i \phi_i \tag{2.2}$$

where Σ_x is the covariance matrix, and λ_i the Eigenvalue associated with the Eigenvector ϕ_i[11]. Basically, this transformation enables the data to be expressed in terms of the patterns between them, instead of the original axes. Furthermore, the Eigenvectors with (normalized) higher Eigenvalues cover more variance of the original data, which means we can remove the Eigenvectors with low Eigenvalues, and thus reducing the dimension of the data, without loosing too much information. If the patterns (dimensions, acoustic features in our case) are highly correlated, a smaller number of Eigenvectors will have large Eigenvalues, hence f will be much smaller than d, and consequently, a large reduction of dimension can be obtained.

PCA can also be interpreted as a probabilistic function (Collins et al., 2001). Each point can be thought of as random number drawn from an unknown distribution P_θ. P denotes a normal distribution with mean θ. The original points (say $x_1, x_2, ...x_n$) are considered to be *noise-corrupted* versions of some true points (say $\theta_1, \theta_2, ...\theta_n$) that lay on a low–dimensional subspace. The aim is to find the latter points, assuming that the noise is Gaussian. The main shortcoming of PCA is the linearity of the model. It assumes that the data follows a Gaussian distribution, however there are a number of situations where the data might follow other type of exponential or non-linear distributions (Collins et al., 2001).

[11] The Eigenvectors of a matrix have the property of being orthonormal.

2.3.3. Labeled data

Collecting a Ground Truth for machine learning is a crucial step towards having a more or less successful algorithm[12]. The Ground Truth is composed of audio related features, extracted automatically from the music piece, and a set of semantic meta-data (tags) related to this music.

As mentioned in Section 2.2, tags for music can be collected in 4 different ways. None of these approaches seems scalable to handle large music collections. However, they can be useful for the creation of a Ground Truth dataset for training and evaluating an autotagging algorithm. As Bertin-Mahieux et al. (2010) mention, a labeled dataset can vary in quality and size. It can be either small and clean (e.g., surveys) or large and noisy (social tags, web documents). Choosing a specific dataset will depend on the type of application (e.g., music recommendation vs. classification) and the type of algorithm (e.g., neural networks vs. SVM).

For instance, the CAL500 dataset, a small and clean dataset obtained from a thorough survey conducted by the CAL laboratory in San Diego[13], has been extensively used in the last few years (Bertin-Mahieux et al., 2008; Hoffman et al., 2009; Marques et al., 2011; Ness et al., 2009; Panagakis et al., 2010; Seyerlehner et al., 2010; Turnbull et al., 2008b). Bertin-Mahieux et al. (2008) and Eck et al. (2008) collected tags from Last.fm[14], a music social website which allows users to scrobble music they are listening to, as well as the ability to add tags to their music, either for artists, tracks or albums. Mandel & Ellis (2005), on the other hand, used the uspop2002 collection (Ellis et al., 2003), a dataset of 8764 tracks with corresponding styles as meta–data, retrieved from the expert music web site AllMusicGuide[15]. The AllMusicGuide web site was also used by Turnbull et al. (2006) in previous experiments, where the authors crawled album reviews and extracted words based on predefined vocabulary of musical words. Magnatagatune[16], which consists of human annotations from the TagAtune game (Law et al., 2007) and audio clips from the DRM–free Creative Common licensed music website Magnatune[17], has been used by a number of publications related to audio tag classification (Hamel et al., 2011; Marques et al., 2011; Ness et al., 2009). More recently, Bertin-Mahieux et al. (2011) published the Million song dataset, "*a freely-available collection of audio features and metadata for a million contemporary popular music tracks*".

[12] the success of an autotagging algorithm will also depend on the type of application the algorithm is built for (Bertin-Mahieux et al., 2010).

[13] http://cosmal.ucsd.edu/cal/

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

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

[16] http://tagatune.org/Magnatagatune.html

[17] http://magnatune.com/

2.3.4. Machine learning

Broadly speaking, Machine Learning is a process of making computers learn or improve/optimize models, by using observations (example data) or past experience. The resulting models can be predictive, that is, they can make future predictions; or descriptive, obtaining more knowledge about the data at hand. Machine Learning is mainly used in complex problems which can be resolved by humans, but either they cannot explain exactly how they solve them, or the process of solving the problems is very time and cost consuming, Basically, we want computers to learn rules (models) which will help us decide whether a future sample belongs to a certain class (or multiple classes) or not, or even to how much degree does the sample belong to the class. This class of problems is known as *Supervised Learning*, since we are supervising the algorithm on the classes to be learned, as opposed to *Unsupervised Learning*, where we want algorithms to learn regularities in the data, if there are any.
In Supervised Learning, the problem is narrowed down to solving the following equation:

$$y = g(x|\theta) \tag{2.3}$$

where $g(\cdot)$ is the model with parameters θ to be solved; x is the input and y is the output. The value of the output y can be a discrete value, if we are considering *classification*, or a continuous function of real–valued elements, in the case of *regression*,
Nowadays, several supervised algorithms have been proposed, differing in the way they learn models in order to discriminate between different classes. Some methods proceed by estimating the densities of the data distribution inside class regions (Alpaydin, 2004). which will give a series of discriminant functions. This kind of methods are known as *Likelihood–based Classification*. Other methods, often called Discriminant–based Classification, may decide to bypass the estimation of those densities and focus directly on the discriminants/boundaries between the class regions.
In the field of music tagging, which started as a generalization of previous MIR tasks such as genre or mood classification, we define the input, x, as a series of acoustic features extracted from the audio and usually presented as a bag of features (unordered in time, or averages and variances of the whole audio piece). The output, y, can be any word defined by the vocabulary.
In the remainder of this subsection, we briefly describe some of the different types of Machine Learning algorithms, and how they are used or modified by the state of the art.

Likelihood–based Classification. As we briefly mentioned before, Supervised Learning algorithms learn models given by Equation 2.3. y takes a discrete value in a classification task or a continuous real–valued function in regression. The model $g(\cdot)$ to be learned is thus a discriminant function or a

regression function, respectively. The Machine Learning algorithm optimizes parameters θ such that the approximation error to the training set is minimized (Alpaydin, 2004).
Depending if we make an assumption on the structure of the data distribution and the variables that define it, we can classify Likelihood–based Classification methods in *parametric*, *semi–parametric* and *non–parametric*.

Parametric methods. If we assume there exists a model valid over the whole input space (a same linear function in regression), all samples of a class are drawn from the same density. Therefore, the goal reduces to finding only the parameters θ to solve the problem. The most prominent density distribution function in statistics and Machine Learning is the *Gaussian* or *Normal* distribution. Given Equation 2.3, if we assume x normal distributed with mean μ and variance σ^2, the density is defined as:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]} \tag{2.4}$$

if the input is univariate, that is, x is a unique input or as:

$$p(x) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}} e^{\left[\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}\right]} \tag{2.5}$$

if the input is multivariate, that is, x is d–dimensional and normal distributed ($x \sim \mathcal{N}_d(\mu, \Sigma)$). In the latter case, μ is the mean vector and Σ is the covariance matrix. The main asset of parametric methods is that models are defined with just few parameters. For instance, in a Gaussian distribution we only need to estimate two parameters, μ and σ^2, in the case of univariate classification, or a mean vector and covariance matrix in multivariate classification. Once these parameters are estimated, the model is known (Alpaydin, 2004).
The aim then is to estimate the density function $p(x)$. In classification, the densities are the different classes (e.g., genres, tags) to learn, denoted by C_i. Therefore, the method estimates $p(x|C_i)$ and priors $P(C_i)$, by Maximum Likelihood Estimation (MLE), in order to compute posteriors $P(C_i|x)$, by using the Bayes' rule, and classify samples (Jaynes & Bretthorst, 2003). Given a set of independent and identically distributed samples $X = \{x^t\}_{t=1}^N$, and assuming that x^t are drawn from the same probability distribution family, $p(x|\theta)$, the Maximum Likelihood Estimation method tries to find θ that makes sampling x^t from $p(x|\theta)$ more likely.
When estimating the covariance matrix in multivariate classification using a Gaussian model, we might end up with a unstable covariance, specially if we have few samples in the training set. In these cases, it is encouraged to reduce the dimension of the input.
Dimension reduction can be achieved in two different ways:

- by performing feature selection, that is, picking a subset of features s from the original features d, where $s < d$, that correlate well with the classes, but have lower inter-correlation.
- by combining the original d dimensions to a reduced $k < d$ dimensions. Typical feature combination algorithms (also known as feature extraction) include Principal Components Analysis (PCA), which was described in Section 2.3.2, Linear Discriminant Analysis (LDA), Relevant Component Analysis (RCA) and Non-negative Matrix Factorization (NMF).

Semi–parametric methods. Most of the times, we cannot make an assumption that the data is drawn from a known density distribution (e.g, normally distributed). Instead, we tend to use a mixture of distributions to estimate the input sample. The mixture density is given by:

$$p(x) = \sum_{i=1}^{k} p(x|G_i)P(G_i) \tag{2.6}$$

where G_i is a mixture component and k (specified beforehand) is the number of mixtures. Such methods are commonly known as semi–parametric, since they are run in two steps. In the first step, the algorithm learns each component density and proportion. In this step, we do not know which observation belongs to which component. Hence, we are facing an unsupervised learning problem. Estimating the mixture components is fulfilled by using an algorithm such as K–means, or more significantly, Expectation Maximization (EM). Once the components are estimated, the second step considers each class as a mixture of a given number of components.

$$p(x|C_i) = \sum_{j=1}^{k_i} p(x|G_{ij})P(G_{ij}) \tag{2.7}$$

In other words, as Alpaydin (2004) states, preceding a supervised learner with unsupervised clustering means that we first learn what normally happens in the data, and then learn what that means.

Non–parametric methods. Another approach to estimating an output for a given input is to consider that the data speaks for itself. In other words, similar inputs have similar outputs. There is no need to learn parameters to fit a global distribution model. Given an instance sample, a non–parametric method attempts to find similar past samples from the training set using a certain distance measure and interpolate from it to find the suitable output. Non–parametric methods work on local models (similar instances) rather than a single global model. These methods are also known as Lazy learning methods, since, unlike parametric methods, they do not need to compute the models beforehand, but rather postpone the computation until they are presented with a test instance. The operation of finding similar instances is of order $O(N)$ (where N is the number of samples in the training set), which is significantly higher than parametric approaches (of order $O(d)$ or $O(d^2)$). Additionally, in parametric approaches the training is done once for the whole test set, whilst the non–parametric requires finding similar instances for each sample in the test set. To overcome this problem, techniques such as dimension reduction (e.g., PCA) are used to reduce the feature space while still keeping the structure of the data. Examples of non–parametric algorithms include Histograms, Kernels, and more specially K–Nearest Neighbors.

k–Nearest Neighbor. The k–Nearest Neighbor (from now on k–NN) classifier is a non–parametric method which, given an input sample, assigns a class having most examples among the k neighbors of the input. The samples are represented as points in a d–dimensional space. (Alpaydin, 2004; Cover & Hart, 1967) The k–NN density estimate is given by:

$$\hat{p}(x) = \frac{k}{NV^k(x)} \tag{2.8}$$

where k is a fixed number of nearest neighbors, N is the total number of instances in the "training" dataset, and $V^k(x)$ is the volume of the d–dimensional hypersphere centered at x, with radius $\left\|x - x_{(k)}\right\|$, being $x_{(k)}$ the k–th nearest observation to x. The k–NN density estimate conditioned to a class of instances is represented as:

$$\hat{p}(x|C_i) = \frac{k_i}{N_i V^k(x)} \tag{2.9}$$

where k_i is the number of nearest neighbors out of the k nearest neighbors that belong to class C_i, and N_i is the number of instances of class C_i in the whole dataset. We can then classify an input instance by solving the following Bayes rule:

$$\hat{p}(C_i|x) = \frac{\hat{p}(x|C_i)\hat{P}(C_i)}{\hat{p}(x)} \tag{2.10}$$

The Maximum Likelihood Estimate (MLE) of the prior density $\hat{P}(C_i)$ is given by:

$$\hat{p}(C_i) = \frac{N_i}{N} \tag{2.11}$$

Substituting equations (2.8), (2.9) and (2.11) into (2.10) yields:

$$\hat{p}(C_i|x) = \frac{\frac{k_i}{\cancel{N_i}\cancel{V^k(x)}} \frac{\cancel{N_i}}{\cancel{N}}}{\frac{k}{\cancel{N}\cancel{V^k(x)}}} = \frac{k_i}{k} \tag{2.12}$$

All neighbors have equal votes, and the class is decided by averaging the number of votes obtained from the k nearest neighbors. In the case of music autotagging, where each song can be annotated with different and not mutually exclusive tags, the algorithm assigns a number of tags having a given threshold of examples/votes among the k nearest neighbors.

Two main problems arise when using a non–parametric algorithm, such as k–NN, for estimating densities in high dimensional spaces (in our case, d is in the order of hundreds). First, the cost/complexity of the algorithm is of order $O(N)$[18], which can make the problem computationally unfeasible if the dimension (d) is high. And second, the concept of closeness (for example, an Euclidean distance between two points) becomes more and more unclear when the dimension increases, the so-called curse of dimensionality (Aggarwal, 2005; Beyer et al., 1999; Korn et al., 2001). One way of dealing with both shortcomings is to reduce the dimension of the data, while still preserving the "essential" information.

Discriminant–based Classification. Whilst Likelihood–based Classification requires estimating densities of class regions, Discriminant–based Classification focus instead on the correct estimation of the boundaries between the class regions. Interestingly, Cortes & Vapnik (1995) state that estimating discriminants (boundaries) is an easier task estimating the densities. This theory only holds if the discriminants can be approximated by a simple function, e.g., linear functions.

The most prominent algorithm for finding discriminants is the Support Vector Machines (SVM) algorithm, proposed by Cortes & Vapnik in 1995 , where basically we need to learn an optimal separating hyperplane, between instances belonging to one class and instances not belonging to that class. Support Vector Machines is thus a supervised binary linear classification algorithm.

If the problem to be solved is not linear, instead of using a non–linear method to fit the discriminant, we apply some non–linear transformations, making use of basis functions (Kernel functions in the case of SVM), to map the problem to a new space and then employ a linear model on this new space, hence reducing considerably the complexity of the problem.

Boosting. Boosting is not exactly a Machine Learning algorithm, but rather a meta–algorithm built upon a set of weak classifiers[19], which, after an iterative process of performing weak learning, re–weighting the data and focusing more

[18] N is the total number of instances.

[19] classifiers that produce a hypothesis which is only slightly better than random guessing.

on the misclassified instances, will produce at the end a strong classifier that is more correlated to the correct classification, in other words, it has an arbitrarily low error ratio of misclassifying instances. The most significant proposed boosting algorithm that adjusts to the errors of weak learning is AdaBoost, published by Freund & Schapire (1997).
From a total of $t = 1, 2, ...T$ iterations, specified beforehand, AdaBoost calls a weak learner iteratively with a distribution $p^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t}$, where w is a weight vector drawn initially from the distribution and N is the number of labeled samples. The weak learner returns a hypothesis h_t, that has an error ε_t. This error rate is used to re–weight the vector w, by increasing the weights of misclassified instances, or alternatively by decreasing the weight of correctly classified instances, therefore the new learner focuses more on the misclassified examples.

2.4. Evaluation

Once a classifier is trained, different methods to evaluate its performance can be used. This section presents some of the methods that are used to evaluate the performance of a classifier. The section in divided in two parts. Section 2.4.1 describes the evaluation measures used to compare the outputs obtained with a classifier to the original data. Usually, classifiers are trained and tested with the same data sets. This technique, however, reduces the generality of the classifier and does not provide any idea on the behavior of the trained classifier in front of new data. For that reason, Section 2.4.2 presents some well known techniques in statistics and machine learning that enable the use of the same data set for training and for testing an algorithm.

2.4.1. Evaluation measures

In this subsection, we introduce the measures used throughout the evaluation of our autotagging algorithm. These measures have been adopted in a variety of fields such as Information Retrieval, Data Mining or Machine Learning. We consider the evaluation of our autotagging algorithm in two different but related tasks: annotation and retrieval. Each task has a different set of evaluation measures.

Annotation

The annotation task can be regarded as a multiclass classification problem where, given a non annotated song, the autotagging system proposes a set of tags, based on an average vote of tags from the most similar songs[20] in

[20] neighbors if we are considering K–NN as a classification algorithm.

the annotated dataset. Therefore, evaluating annotation means evaluating the relevance (quality) and the ratio (quantity) of the obtained tags.
Based on the table of truthfulness/falseness of the null hypothesis in hypothesis testing (Lehmann & Romano, 2005), the so–called Type I and Type II errors, Table 2.3 shows the contingency table of the terms True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), which compare the predicted class (classes in multiclass classification) of an item with the actual class.

Table 2.3: Contingency Table of truthfulness/falseness.

		Actual Class	
		True	False
Predicted Class	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Precision. measures the ratio of predicted classes that are relevant. The equation is given by:

$$P = \frac{TP}{TP + FP} \tag{2.13}$$

Recall. , on the other hand, measures the ratio of relevant classes that were predicted. The equation is:

$$R = \frac{TP}{TP + FN} \tag{2.14}$$

F–measure. , or F_1–score, is a weighted harmonic mean average measure of both precision and recall:

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = 2\frac{PR}{P + R} \tag{2.15}$$

Alternatively, we can define different values for the F_β, to put more emphasis on either precision or recall. The general formula of the F_β–measure for a given positive real number β is:

$$F_\beta = \frac{1 + \beta^2}{\frac{1}{P} + \frac{\beta^2}{R}} = (1 + \beta^2)\frac{PR}{(\beta^2 P) + R} \tag{2.16}$$

For example, when $\beta = 2$, we are putting more emphasis on recall than precision.

Spearman's rank correlation coefficient. Sometimes, we want to evaluate the effectiveness of our system beyond the binary decisions of predicted or not predicted. Our algorithm, by definition, proposes tags with a corresponding frequency for each tag. This frequency is based on the weighted votes among the similar songs to a given one. Neither precision nor recall (and consequently neither *f–measure*) take into account the frequencies (i.e. ranking) of the tags obtained from the similar songs. Thus, we used the Spearman's rank correlation coefficient, or Spearman ρ, which is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.17}$$

Where d_i represents the distance between each rank of pair of values —in our case labels in the ground truth and labels in the proposed tags— and n the number of all possible pair of values. To compute the distances, if it is not explicitly set, we assume that the frequency of manually annotated labels is equal to 1.

Retrieval

In the task of music retrieval, we want to emulate the functionality of a search engine. Given a query tag, we want to measure how well our system is able to return relevant songs. The retrieved songs are normally ranked/ordered based on the relevance to the given query tag. As mentioned in the annotation task, precision and recall (and consequently *f–measure*) are suitable for unordered sets. Thus, we need to refine these measures or define new ones for ordered sets. We described *Spearman rho* in the annotation task. In the case of music retrieval, we use the following evaluation measures.

Mean Average Precision. Average Precision, for a given query tag q_j, is the average precision measure of the top n documents after each relevant document is found, as we are moving down through the ranked list. The formula of Average Precision is given by:

$$AveP_j = \frac{1}{M_j} \sum_{n=1}^{M_j} precision(R_{jn}) \tag{2.18}$$

where M_j is the number of relevant songs for query tag q_j, and R_{jn} is the set of ranked retrieved songs from the top until we get a relevant song s_n. The Mean Average Precision, or MAP, is the arithmetic mean of the Average Precision for all tags in the test set. That is:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} AveP_j \tag{2.19}$$

where Q is a set of tag queries $\{q_1, q_2, ..., q_j\}$. MAP can be regarded (roughly/approximately) as the area under the Precision–Recall curve.

Precision–at–N. Traditional text–based search engines (due to screen limitations) tend to return a limited number of documents in the first page of results. This value is normally 10 or 30. In such cases, we want to measure the precision at a fixed level of retrieved results.
Thus, we use the Precision–at–N measure, which is a modification of the plain Precision measure. If the Precision for retrieval is defined as:

$$P = \frac{\#relevant\ songs\ retrieved}{\#retrieved\ items} \tag{2.20}$$

The Precision–at–N, or $P@n$, is:

$$P@n = \frac{\#relevant\ songs\ retrieved\ out\ of\ n}{n} \tag{2.21}$$

This measure is very useful if we do not know a priori the exact number of relevant songs to a given tag. However, it is strongly affected by the number of relevant songs for a given query tag q_j.

R–Precision. R–Precision is a related measure that mitigates the problem of Precision–at–N, by defining a priori a set of R relevant songs for a tag query q_j[21].

$$R\text{–}Precision = \frac{\#relevant\ songs\ retrieved}{R} \tag{2.22}$$

Area under the ROC curve. As the Mean Average Precision is roughly the area under the precision–recall curve, the Area under the ROC curve, or AUC, as its name suggests, is the area under the Receiver Operating Characteristic curve. Basically, the ROC curve is the plot of the rate of false positive rates (also known as 1–specificity) and true positives (sensitivity), as we are moving down through the ranked list of retrieved songs. The True Positive Rate (TPR) and False Positive Rate (FPR) are defined as:

$$TPR = \frac{TP}{TP + FN} = Recall \tag{2.23}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.24}$$

2.4.2. Validation

The following is a list of techniques that are used to allow training and testing a classifier with the same dataset.

[21] this list can be complete or incomplete.

v–fold cross–validation. The original dataset is splitted into v equally distributed and mutually exclusive subsamples. Then, a single subsample is retained as the test set, and the remaining $v - 1$ subsamples are used as training data. This process is repeated v times (the folds), with each of the v subsamples used exactly once as the test set. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation.

Leave-one-out. This technique uses a unique observation from the original dataset to validate the classifier, and the remaining observations as the training data. This process is repeated until each sample in the original dataset is used once as the test set. This process is similar than v–fold cross–validation but setting v as the number of observations in the original dataset.

Holdout. This method reserves a certain number of samples for testing and uses the remainder for training. In other words, it is equivalent to randomly splitting the dataset into two subsets: one for training and the other for testing. It is common to hold out one-third of the data for testing. From the conceptual point of view, Holdout validation is not cross-validation in the common sense, because the data is never crossed over.

Bootstrap. This technique estimates the sampling distribution of an estimator by sampling the original sample with replacement with the purpose of deriving robust estimates of standard errors of a population parameter (mean, median, correlation coefficient, etc.).

The Leave-one-out method tends to include unnecessary components in the model, and has been provided to be asymptotically incorrect (Stone, 1977). Furthermore, the method does not work well for data with strong clusterization (Eriksson et al., 2000) and underestimates the true predictive error (Martens & Dardenne, 1998). Compared to Holdout, cross-validation is markedly superior for small data sets (Goutte, 1997);

2.5. Related work

In the last few years, several algorithms have been proposed for the task of automatic tagging of music. All these algorithms are instances or modifications of the algorithms proposed in related classification tasks . Most of these algorithms are based on previous algorithms proposed and used in other related Music Information Retrieval (from now on MIR) research areas, such as genre, mood, instrumentation or even sound classification (Cano & Koppenberger, 2004; Cano et al., 2005; Herrera-Boyer et al., 2003; Liu et al., 2003; Logan, 2000; Tzanetakis & Cook, 2002).

Although not specifically oriented to tag classification, the work of Whitman et al. (2004; 2002; 2002) is considered one of the first referenced works in music semantic annotation/description literature. Whitman (2005) attempts to learn the relationship of text found in web documents (such as album/record reviews (Whitman & Ellis, 2004)) and the actual acoustic description of a music piece. They accomplish so by using Power Spectrum Density over one second of audio, and then reducing the dimension of the data by means of Principle Component Analysis (PCA), to a final 20 dimensions. Thereafter, an algorithm based on Regularized Least Squares Classification (RLSC) (Rifkin et al., 2003) is used. This algorithm is similar to SVM in that it is also a linear (discriminant–based) algorithm. However, unlike the SVM algorithm, which requires the solution to a convex quadratic programming problem, training a RSLC algorithm only requires solving a single system of linear equations (Rifkin et al., 2003).

On the other hand, a considerable amount of work in classification has been published in the sound effects domain. For instance, Slaney (2002) addressed the problem of audio retrieval by creating separate hierarchical models in the test and the audio space, and then making links between the two spaces for either retrieval or annotation. Similarly, Cano & Koppenberger (2004) propose a sound effects annotation approach based on nearest neighbor classification. In other related work, Cano et al. (2005) used an extended WordNet version, and a one nearest neighbor decision rule to classify sound effects. Both studies used a repository that centralizes audio content, corresponding meta data, taxonomies and algorithms (Cano et al., 2004a). In fact, our autotagging algorithm builds upon the previous findings of Cano & Koppenberger.

Mandel & Ellis (2005) proposed the use of SVM's for automatically tagging music. As mentioned in Section 2.3.4, the SVM algorithm attempts to find the optimal margin hyperplane to separate two different classes, in terms of the distance of the closest points from each class to that hyperplane. If the data is not linear (which happens most of the times), it is projected to a higher dimensional space using Kernel functions. Mandel and Ellis suggested the use of 3 different distance measurements: 1) Mahalanobis distance, 2) Kullback–Leibler divergence (approximated by Monte Carlo methods) over 1 or 3) 20 gaussian Gaussian components (GMM's), which were trained from 3000 MFCC frames of the whole music piece (representing 10–20% of the total frames per song).

Turnbull et al. (2007a), based on the work by Carneiro & Vasconcelos (2005) in the image annotation domain, proposed to view the problem of semantic annotation (autotagging) as a Multi–class classification problem rather than a one–versus–all binary problem. According to the authors, the advantage of such an approach is that, since one–vs–all approaches require the use of negative examples, in a weekly labeled training dataset, if a word does not appear in the list of tags of a given song, it does not necessarily mean that the song cannot be annotated with that word. The distribution to fit in their case is given by:

$$p(X|i) = \sum_{c=1}^{C} \pi_c \mathcal{N}(x|\mu_c, \Sigma_c) \tag{2.25}$$

where $\mathcal{N}(\cdot)$ is a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. In order to avoid overfitting of the data, the authors use diagonal covariance matrices. The parameters are then estimated using the Expectation Maximization (EM) algorithm. The estimation is done directly over the word–level distribution, by taking a reduced set of random samples, or by first estimating song–level distribution and then averaging. More recently (Turnbull et al., 2008b), the authors suggested a weighted Mixture Hierarchies estimation, which is computationally much less expensive than model averaging, while still using all the training data without sub-sampling.

Eck et al. (2008), used the AdaBoost algorithm to automatically tag music. Following the same idea, Bertin-Mahieux et al. (2008) used FilterBoost, an on-line version of the former, which employs rejection sampling with the result that it can handle sampling of arbitrarily large music data sets.

More recently, Hoffman et al. (2009) proposed a probabilistic model, called Code Bernoulli Average (CBA). Unlike other probabilistic methods, which assume that songs (data) depend on the tags (classes), the CBA algorithm attempts to predict to how much degree does a tag apply to a song, based on a reduced, Vector Quantization representation of that song. The authors first discretize the data representation by using Vector Quantization[22]. Then, the model parameters of the CBA are estimated with Maximum Likelihood estimation using the EM algorithm.

In the last few years, there has a been a growing interest in the use of two–stage algorithms. Broadly speaking, a two–stage algorithm uses the output of a content–based algorithm as input feature vectors to model each tag in the vocabulary. The rationale behind two–stage algorithms is that they explicitly tackle the problem of tag correlation (Aucouturier et al., 2007b). A number of authors report on the performance improvements using this method (e.g. Coviello et al., 2011; Miotto et al., 2010; Ness et al., 2009; Pachet & Roy, 2009).

For instance Aucouturier et al. (2007b) presented a hybrid classification algorithm, consisting of 2 components. The first component is an acoustic classifier which models acoustic features with 50–state Gaussian Mixture Models (GMM) and uses K–NN with Kulback Leibler divergence as a distance measure. The idea is that similar songs are supposed to lie on a similar space. The second component is a set of decision–tree classifiers which exploit symbolic–level correlations between meta-data to improve previous classifications. This second component is an iterative procedure that will stop when there is no more significant improvements of precision between successive iteration[23].

[22] using K–means as a method for dimension reduction.

[23] each iteration corresponds to a decision–tree classifier instance.

CHAPTER 3

Automatic annotation of music from audio

3.1. Introduction

Automatic tagging of music, or simply music autotagging, refers to the task of classifying audio items in terms of high-level concepts, such as `musical genre`, `instrumentation`, `moods`, etc.
As it has been presented in previous chapters, most of the autotagging approaches are extensions of previous work in related tasks, including genre, mood and artist classification. Classification is carried out by learning models automatically from the mapping between low level audio features and semantic labels, or tags. Nevertheless, as pointed out recently by Marques et al. (2011), the task at hand is much more difficult than genre or mood classification, for several reasons. We try to summarize these differences as follows:

- The number of tags to classify are in the order of hundreds, and it is much higher than genre or mood classification.

- The number of facets (categories) is also higher. Genre and mood are only two facets among many others.

- Some of these tags are subjetive or not clearly linked with the audio content. For example, usage tags such as *cleaning the house* or *driving* might have different meanings for different people. Some personal tags are very hard to classify, like *favorite* or *seen live*.

- Autotagging is a multiclass problem. An audio excerpt is usually annotated with more than one tag.

- Tags are often correlated. That is, two or more tags can share concepts (e.g., *rock* music and *guitar* playing), they can be synonyms (e.g., a *vigorous* or *energetic* performance).

- Tags can have multiple meanings (polysemy). For example, the tag *piano* might refer to the instrument or the tempo.

- Because of the multiplicity of potential relevant facets and tags, available datasets are often *weakly labeled*, i.e. the absence of a particular tag annotation for a given excerpt does not necessarily mean that the tag is not relevant to the excerpt.

- For the same reason, available datasets are often noisy and inconsistent.

In this chapter, we introduce our automatic music tagging approach[1]. Our algorithm proposes tags from already annotated songs that are acoustically similar to the given one, as opposed to many other approaches that learn models or discriminants from the observations beforehand (as in the case of parametric and semi-parametric methods, such as Gaussian Mixture Models, Boosting methods or Support Vector Machines). That is to say, our algorithm propagates tags to an unlabeled song, say s, from the k nearest neighbors in the "acoustic space".

Our approach is built upon previous work by Cano & Koppenberger (2004), which automatically annotate sound effects using nearest neighbor classification. However, our work differs on the way similarity distance is calculated, and all the heuristics and parameters that are constructed on top of the nearest neighbor classifier.

Figure 3.1 depicts the general framework of our proposed autotagging algorithm. The raw audio files of the Ground Truth training dataset are analyzed by extracting acoustic features and performing a feature selection. Classification is then achieved by applying the same set of transformations for each test song, and then using similarity distances to infer tags from neighbors in the training dataset. In other words, given a seed song, our audio analysis module extracts the acoustic features and performs the same feature selection as the Ground Truth training dataset. The resulting output of the audio analysis module, a d–dimensional feature vector, is queried into the autotagging module. This module returns a list of k nearest neighbors of s in the projected d–dimensional space. Then, based on a voting function of the neighbors' tags, the autotagging module proposes a list of tags to the query song s, where each tag has a corresponding frequency/weight given by the voting method.

As mentioned in Section 2.3.3, a Ground Truth dataset can vary in size and quality of the data, depending on the methodology used to collect the data. Ground Truth tags can be obtained in four different ways: conducting human surveys[2], deploying games (with a purpose), collecting web documents or harvesting social tags (Turnbull et al., 2008a). In order to assess the strength

[1] From this point now on, the concept of automatically classifying an audio with tags, or automatic tagging, will be simply referred as "autotagging".

[2] hiring musical experts to annotate music can be regarded as a special case of human surveys.

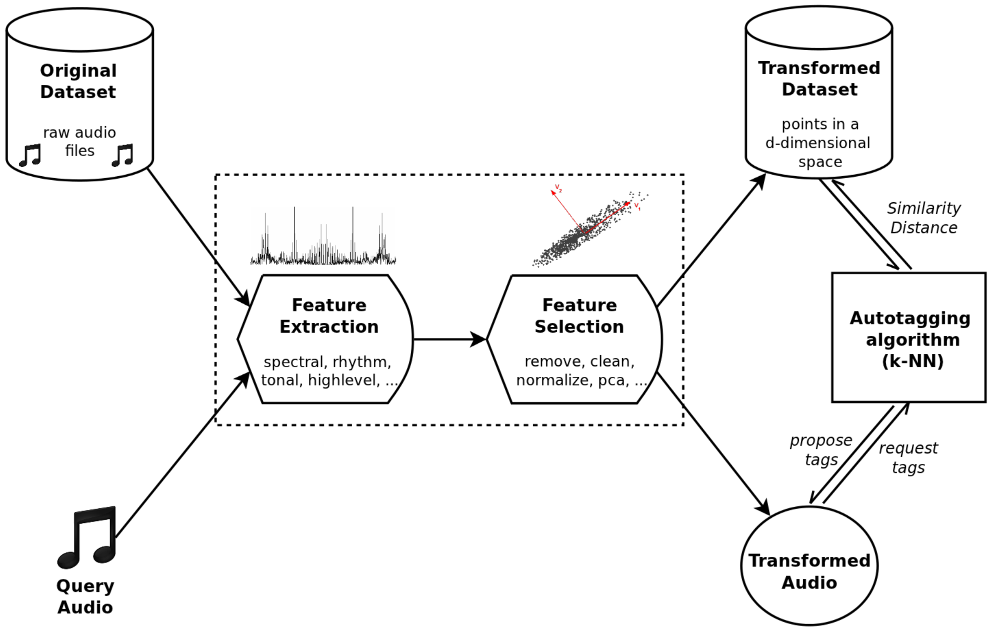

Figure 3.1: Proposed general framework for automatic tagging of music.

of our autotagging algorithm, we carry out a thorough evaluation using six datasets from different sources.

In the remainder of this chapter, we focus on the different aspects of our autotagging algorithm, following the diagram depicted in Figure 3.1. Section 3.2 describes our proposed autotagging algorithm. We provide details on the feature extraction, as well as the feature selection process that was followed to reduce the dimension of the audio data. Furthermore, we describe the learning algorithm used, and the parameters that can be tuned to modify the performance of our algorithm. Section 3.3 presents the first experimental results —as a proof of concepts— of our algorithm. The aim of Section 3.4, is to analyze if our approach can also apply to sound effects. In Section 3.5 we run statistical tests to assess whether different parameter selection can significantly affect the performance of our algorithm. Then, in Sections 3.5, 3.6 and 3.7 we report on the evaluation of our system in different experiments, using the output of the statistical tests in Section 3.5 as a parameter configuration for our approach. Finally, we conclude in Section 3.8.

3.2. Autotagging algorithm description

3.2.1. Feature extraction

The most prominent audio features in MIR tasks such as audio genre classification and audio tag classification, are the Mel Frequency Cepstral Coefficients (MFCC's). In our case, we use the first 13 coefficients as one of many other sets of features (descriptors), ranging from low–level spectral features (such as centroid, rolloff, kurtosis, etc.), to tempora/rhythm (bpm, onset), tonal descriptors (chords, key) and more high–level descriptors (moods, genres, danceability,...). The latter features are learned — in a first stage — as probability estimations made by Support Vector Machines using a set of internal Ground Truth datasets. For more details about these high level features please refer to Bogdanov et al. (2011).
Table 3.1 summarizes the list of features that are used by our autotagging algorithm, which were extracted with the `Essentia` library (Wack, 2011). The audio features are captured on a short-time frame-by-frame basis, using sliding windows of 46ms, and a hop size of 23ms. For tonal features, we set these values to 92ms and 46 ms, respectively. The decision of choosing a bigger window for tonal features — such as pitch class profiles — is based on two reasons. First, the feature tries to model the time required by humans to identify a chord. And second, it is necessary to get good frequency resolution, thus a large frame is required (Fujishima, 1999; Gomez, 2006). The features are then averaged over the whole audio excerpt (Tzanetakis & Cook, 2002). We take the means, variances and their corresponding derivatives. These values are then used to represent each audio excerpt as an N–dimensional vector.
Then, depending on the parameter configuration (refer to Section 3.2.4) and the distance measure we deploy in our k–NN implementation, we use different subsets of these features.

3.2.2. Feature selection

Machine learning algorithms learn and optimize models from previous observations of data, in order to make future predictions of unseen events. In the Music Information Retrieval field, an observation usually consists of different audio features, extracted from the music excerpt in the time and frequency domain. These features are normally extracted from small windows. Consider, for example, a 5 minute song. A single (real valued or string) feature computed from a sliding window of 46ms and a hop size of 23ms will result in a $\frac{5*60*1000-1}{23} \simeq 13043$ dimensional vector. If there are 60 features, a song will be represented by 60 vectors of 13043 dimensions each. The main problems of using such amount of information for a single music item are the following:

- Songs of different lengths will consequently have feature vectors of different length.

Table 3.1: Summary list of the acoustic features used by our autotagging algorithm.

Low level	* average loudness * bark bands (bands, spread, skewness, kurtosis) * dissonance * High Frequency Content * MFCC * pitch (pitch, salience, instantaneous confidence) * silence rate (at 20dB, 30db and 60dB) * spectral (centroid, complexity, contrast, crest, decrease energy, energy band, flatness db, flux, kurtosis, rms, rolloff, skewness, spread, strong peak) * zero crossing rate
Rhythm	* beats (position, loudness, loudness bass) * bpm (bpm, confidence, estimates, intervals) * first peak (bpm, spread, weight) * second peak (bpm, spread, weight) * onset (rate, times) * rubato (start, stop)
Tonal	* chords (key, scale, progression, histogram, strength, changes rate, number rate) * key (key, scale, strength) * hpcp * tuning (frequency, diatonic strength, non-tempered energy ratio, equal tempered deviation)
High level	* genres (from 5 different collections) * moods * gender * western/non-western * live/studio * speech/music * rhythm (fast, medium or slow) * timbre (bright or dark) * voice/instrumental

- As we mentioned in Section 2.3.4, in parametric models, the covariance matrix becomes unstable when the data is described with many dimensions, and there are only few samples from which to build a model.

- As digital collections keep growing at a fast pace, current hardware specifications — although there has been a significant improvement in personal computation — cannot handle such amount of information.

In the last 30 years, many feature selection methods have been proposed for dimension reduction. These techniques are used to remove noisy information from the original feature representation so that learning models becomes a more feasible task (Samet, 2006).

In our case, the following steps are applied for feature selection. For each audio feature[3] we extract the mean, variance and their corresponding derivatives. Then, we concatenate these values into a vector. This reduces the observation to a single vector of 611 dimensions (including high level features). We further reduce the dimension by means of Principal Component Analysis (PCA), after mean centering the 611–dimensional vector. The PCA method can be regarded as a flattening technique (see Section 2.3.2 for more details). It projects the original space into a new subspace, whose basis vectors — also known as components — represent the maximum–variance direction in the original space.

PCA is a non–parametric method, and as such, the dimension reduction — with this technique — is unique and independent of the user. Furthermore, the components of a PCA are, by definition, orthogonal (Jolliffe, 2002). This may have a positive effect, but sometimes it can lead to undesirable effects, especially when the data presents patterns that cannot be described with an orthogonal basis. Many other dimension reduction techniques have been proposed. Some of them follow the same idea of linear combination of variables, such as Linear Discriminant Analysis (LDA) or Non-negative Matrix Factorization (NMF). Others, such as Relevant Components Analysis (RCA) or Independent Component Analysis (ICA) impose constraints on the classes or the dependecy of the reduced dimensions, respectively, and can consequently improve data separability.

Nevertheless, experimental tests in Section 3.5.2 — and subsequent results — show that applying PCA to the original data does not harm the evaluation results significantly[4] while reducing the complexity of the algorithm considerably. For instance, given the 611–dimensional vector mentioned earlier, the PCA projection that keeps the 75% variance of this vector will contain barely 29-30 components (depending on the dataset), that is, a song can be then described by solely a 30–dimensional vector.

3.2.3. Learning algorithm

The purpose of annotating songs with semantic labels, or simply tagging, is manifold. Nevertheless, we can define two basic scenarios where tagging is useful: `music annotation` and `music retrieval`. For the former, given a song, we want the autotagging algorithm to propose a list of tags which is, at the same time, sensitive (recall) and not noisy (precision). For the latter, we

[3] Except for the features with a single value (e.g., key, chords, bpm, etc.) or high-level features, which consist of a string value and a probability value.

[4] by using standard Information Retrieval evaluation measures such as F–measure and Mean Average Precision.

want the autotagging algorithm to propose a list of tags to songs such that, given a query tag —or a list of tags— the retrieved songs are relevant to the query.

In this section we present two versions of our autotagging algorithm. The main approach is to consider the k nearest neighbors of a given song s, assigning to s the labels that these neighbors share. Sometimes, depending on the shape of the dataset (i.e how the points are spread in the d–dimensional space, after extracting and selecting features from the raw audio files), the vocabulary size, and the number of instances per class, the k–NN method may fail to cover all the vocabulary. In order to tackle this issue, we take another approach. Instead of considering songs that "sound similar," the similarity or likelihood is computed from the query song to tag models. A tag model is built from the training songs that are annotated with that tag. We call this approach Class-Based Distance Classification (CBDC). A detailed description and discussion of the two approaches is presented in the next subsections.

It should be noted though that when we generally use the term "our autotagging algorithm," we tend to refer to the first approach, the Weighted vote k–NN. The Class-Based Distance Classification model is mainly used in Section 3.5 to study the effects produced in the evaluation methods by different autotagging paradigms.

Weighted vote k-NN

As we mentioned in Section 2.3.4, a k–NN method for multi-class classification proceeds as follows: given a seed song s, the algorithm assigns the best classes from the k nearest neighbors. These classes are decided by taking into account the votes of all the k neighbors. Algorithm 3.1 summarizes how our weighted vote k-NN algorithm works. First, a set of k nearest neighbor songs is retrieved for song s. Then, the tags of the k nearest neighbors — with their corresponding weight — are merged into a candidate list. Finally, the candidate list is filtered by frequency, producing a reduced list of proposed tags.

For `music annotation`, all the k nearest neighbors have equal vote, that is, $weight(t) = 1$ for all the neighbors' tags. The *threshold* parameter is defined as a cut–off value that limits the number of proposed tags. For example, in a 10–NN classifier, if 4 out of 10 neighbors of song s have the tag *rock*, and 1 neighbor has the tag *punk* , then the algorithm proposes tags *rock* and *punk* to song s with a frequency $\frac{4}{10} = 0.4$ and $\frac{1}{10} = 0.1$, respectively. If the threshold is set to 0.3, the algorithm will only propose tag *rock* to song s.

In the case of `music retrieval`, or `tag ranking`, the concept of voting threshold is removed, since the ranking is done for all the tags in the Ground Truth vocabulary. Instead, we take the $k = \mathcal{R}$–nearest neighbors, where $\mathcal{R}$ is the number of songs in the training dataset, and rank tags based on the following weighting function:

```
Data: $s$, a song represented as a point in a $d$–dimensional space;
$k$, number of nearest neighbors;
$threshold$, a cut-off threshold to limit the number of proposed tags;
Result: $P$, a list of proposed tags;
$T = \emptyset$;
$P = \emptyset$;
$N = k\text{–}Nearestneighbors(s)$;
foreach $n \in N$ do
    foreach $t \in tags(n)$ do
        $T \leftarrow T \cup (t, weight(t, n))$;
    end
end
foreach $t \in T$ do
    if $(freq(t) \geq threshold)^*$ then
        $P \leftarrow P \cup t$;
    end
end
```

Algorithm 3.1: Pseudo-code for the weighted vote k-NN algorithm.* The cut–off threshold (`if` condition) is only defined for music annotation, not for retrieval.

$$weight(t, n) = \begin{cases} 1, & \text{if } n \geq k \\ \frac{1}{n^2}, & \text{otherwise} \end{cases} \tag{3.1}$$

where $weight(t, n)$ is the weight or score of tag t in rank n (n–nearest neighbor). The value of k is taken from the music annotation task. In other words, the first k nearest neighbors will affect the classification equally, whilst the furthest neighbors ($R - k$) are defined by a reciprocal quadratic function. This function is set to give a marginal weight for the furthest neighbors, so the nearest neighbors will have more influence on the highly ranked tags, while still allowing the ranking of all the tags in the Ground Truth vocabulary.

Comments on the algorithm. The choice of a memory–based nearest neighbor classifier avoids the design and training of every possible tag. Furthermore, there is no need to use negative labels (e.g., *rock* and *not rock*), since there are no tag models to be learned. Another advantage of using an NN classifier is that it does not need to be redesigned nor trained whenever a new class of audio excerpts is added to the system. The autotagging problem is then reduced to finding a suitable similarity distance between songs.

Nevertheless, it is well known that the $k-$NN classifier has some limitations, for some of which we tried to find a solution in the case of music tag classification:

- k–NN can be easily misled in high–dimensional spaces. Our feature selection step (using PCA) reduces considerably the number of dimensions while still keeping the essential information of the original data.
- k–NN is a lazy algorithm, that is, it postpones the classification until the test instances are provided. For each test instance, the algorithm has to compute the distance from this instance to all the training instances. Thus, k–NN will require as many distance calculations as the number of instances in the Ground Truth dataset. While we do not face this problem directly —in this algorithm—, computing distances between points in a reduced dimension (In our algorithm, in the order of 7% of the original data representation) can be efficiently computed in current hardware specifications (see Section 3.7.2 for computational costs).

In some situations where the tag frequency is considerably unbalanced and the data representation is not spread —that is, if the songs are not well distributed in the "acoustic space"— the weighted vote k–NN method is not able to cover all the tags while classifying in a cross–validation basis. Since the singularity of a non-parametric method is that data speaks for itself, on one hand, songs that "sound similar" in our data representation but are annotated with different[5] tags, may consequently end up confusing the system. We tackle this problem by using a voting threshold that limits the number of proposed tags. On the other hand, given the aforementioned voting threshold and the nature of the algorithm, tags that are less frequent are then less prone to be proposed. One way to deal with such problem is to disable the voting threshold and reduce the number of nearest neighbors. Experimental results (see Section 3.5.3) show an increment of the covered tags, especially for the tags that are used less frequently in the training dataset. Interestingly, this finding shows an improvement in per–tag evaluations and a decay in per–song evaluations. The increment of covered tags, however, was not enough to cover all the tags in the vocabulary. Based on these findings, we propose the use of an alternative approach, which is aware of the tags of each song in the acoustic space.

Class–based Distance Classifier

The Class–based Distance Classifier (CBDC) is an instance of a centroid-based distance classifier (Han & Karypis, 2000; Kim et al., 2006; Park et al., 2003). It uses the same feature representation as in the previous algorithm. However, instead of looking at the nearest songs, it focuses on the nearest tags (hence, class-based). For each tag t in the vocabulary, we compute a clustered representation of all the songs that are annotated with this tag in the Ground Truth. The process is summarized in Algorithm 3.2.

[5] Here different means tags that differ in their names. Sometimes tags with different names can refer to the same or a related semantic concept, as we are going to see in the following chapters.

```
Data: $s$, a song represented as a point in a $d$–dimensional space;
$GT$, a list of tags in the Ground Truth;
Result: $P$, a list of proposed tags;
$C = \emptyset$;
$P = \emptyset$;
foreach $t \in GT$ do
    $S \leftarrow songs\_with\_tag(t)$;
    $C \leftarrow C \cup centroid(S)$;
end
foreach $c \in C$ do
    $P \leftarrow P \cup (tag(c), distance(s, c))$;
end
$P \leftarrow sorted(P)$;
```

Algorithm 3.2: Pseudo-code for the class-based distance algorithm.

Once the centroids are obtained, the algorithm computes the distance of a test song s to the $|V|$ centroids, where $|V|$ is the size of the Ground Truth vocabulary. We consider four different distance measures: Euclidean, cosine, weighted Euclidean (with a diagonal matrix) and Mahalanobis (with a full covariance matrix) distance.
The generalized Mahalanobis distance (Mahalanobis, 1936) is defined as:

$$d(\overrightarrow{s}, \overrightarrow{c}) = \sqrt{(\overrightarrow{s} - \overrightarrow{c})S^{-1}(\overrightarrow{s} - \overrightarrow{c})} \tag{3.2}$$

where $\overrightarrow{s}$ and $\overrightarrow{c}$ are two vectors corresponding to the test song feature vector and the tag centroid, respectively. S is a covariance matrix obtained from the feature vectors of all the songs that are annotated with the same tag. When S is a diagonal matrix, the resulting distance measure is called `weighted` (or normalized) `Euclidean`:

$$d(\overrightarrow{s}, \overrightarrow{c}) = \sqrt{\sum_{i=1}^{N} \frac{(s_i - c_i)^2}{s_{ii}^2}} \tag{3.3}$$

where s_{ii} is the standard deviation of the test song feature vector ($\overrightarrow{s}$) and the tag centroid ($\overrightarrow{c}$) in the sample distribution. When S is the Identity matrix, then the resulting measure is as easy as a Euclidean distance between vectors $\overrightarrow{s}$ and $\overrightarrow{c}$:

$$d(\overrightarrow{s}, \overrightarrow{c}) = \sqrt{\sum_{i=1}^{N} (s_i - c_i)^2} \tag{3.4}$$

.
Finally, the cosine distance, or cosine similarity, is defined as:

$$d(\overrightarrow{s}, \overrightarrow{c}) = \frac{\overrightarrow{s} \cdot \overrightarrow{c}}{\|\overrightarrow{s}\|\|\overrightarrow{c}\|} = \frac{\sum_{i=1}^{N} s_i \times c_i}{\sqrt{\sum_{i=1}^{N}(s_i)^2} \times \sqrt{\sum_{i=1}^{N}(c_i)^2}} \tag{3.5}$$

The first two distance measures (equations 3.2 and 3.3) can be regarded as a modelization of a single Gaussian for each tag.

Comments on the algorithm. While this approach might seem naïve and generic, we show in experimental results (see Section 3.5.3) that, when the audio feature representation is good enough to differentiate between classes, it can perform — in terms of per–tag evaluations — as well as or better than the state of the art approaches that use more complex, time and resource consuming algorithms. As for the computational cost, the tag centroids, including the covariance matrices, are computed only once, and they scale linearly with the number of tags and the number of instances per tag. Once the centroids are obtained, the classification process is reduced to calculating $|V|$–distances per song, where $|V|$ is the size of the vocabulary, as compared to the $\mathcal{R}$–distances of a k–NN classifier, where $\mathcal{R}$ corresponds to the number of instances in the training dataset, and $|V| < R$. The classification, thus, scales linearly with the number of classes.

3.2.4. Parameter selection

Most machine learning algorithms have a set of parameters[6] that can be tuned to change the performance of the algorithm. In our case, we define five tuning parameters for our main algorithm, the Weighted vote k–NN.

PCA covered variance. With the Principal Component Analysis technique, we want to reduce the dimension of the original data without losing too much information. In other words, keeping as much variance of the original data as possible. In this case we can choose either the number of dimensions (number of eigenvectors) or the amount of variance to keep.

Use of high-level features. As we mentioned in Section 3.2.1, high-level descriptors refer to features such as genre, mood, danceability, etc. learned — in a first stage — as probability estimations made by Support Vector Machines using a set of internal Ground Truth datasets. For more details please refer to Bogdanov et al. (2011).

[6] To not be confused with the parametric–non-parametric types of machine learning algorithms.

Distance metric. Different measures of distance lead to different results. Casey et al. (2008) proposed a set of minimum distances for high-dimensional music spaces. In our case, we use three types of distances: 1) a Euclidean distance between feature vectors in the reduced d-dimensional space, 2) a Kullback-Leibler divergence based on single Gaussian MFCC modeling, and 3) a linear combination of both Euclidean and Kullback-Leibler distances. The Euclidean distance includes the means of the first 13 MFCC coefficients — averaged over the whole audio — as additional features.

Number of nearest neighbors. This parameter affects the value of k in the k-NN algorithm. A lower number means less neighbors, and probably less tags. It depends on how the transformed feature vectors are distributed in the d-dimensional space. If the distribution is spread, a lower number would make the proposed tags more accurate than a higher number. With a higher value, the algorithm can have more tags to choose from, which can increase the probability of having a higher recall.

Voting threshold. The voting threshold parameter affects the number of proposed tags. If the value of this threshold is closer to 1, few tags are propagated, yet we are more confident about them, that is, we increase the probability of having lower recall and higher precision. Otherwise, when the threshold is closer to 0, we are propagating as many tags as all the k neighbors have, which obviously increases the probability of having a higher recall and a lower precision. Empirical results — after the evaluation of our system in all the datasets we used — have shown that a threshold of 0.2 has a good trade-off between precision and recall.

3.3. Experiment 1: Magnatune-5K dataset

3.3.1. Dataset

The first dataset, used in the first experiment as a proof of concepts, consists of more than 5000 songs annotated with style and mood tags. The dataset and the annotations were obtained from the Magnatune website[7] in February, 2007. Magnatune (Buckma, 2004) is a California–based independent music label that offers DRM–free Creative Common licensed music, aiming at treating artists and consumers fairly. The dataset is divided in two subsets (style and moods), corresponding to the two sub-experiments in Section 3.3.2.

The style subset contains 29 different style labels in 5481 annotated songs, with an average of 3.06 tags per song. Table 3.2 shows additional information about the dataset.

[7] http://www.magnatune.com

Table 3.2: Additional information about the Magnatune-5K dataset, gathered from *www.magnatune.com* during February, 2007.

#Tracks	#Tags	Tags/Track	Top Tags	Bottom Tags
5481	29	$\mu = 3.06$	Instrumental (2182)	Blues (118)
		$\sigma = 1.13$	Classical (2175)	Opera (69)
			Baroque (1386)	Indian (60)
			Rock (961)	Other (33)
			World (870)	Children (13)

For the moods experiment, the first issue is the choice of the taxonomy. As advised in (Juslin & Sloboda, 2001), in order to make our experiment and to build a ground truth that achieves the best agreement between people, we should consider few categories. We used a reduced version of the Magnatune on-line library. This collection offers a set of playlists based on mood[8]. We clustered the 150 mood playlists to fit in our few categories paradigm. The adjectives proposed by Juslin: happiness, sadness, anger and fear in (Juslin & Sloboda, 2001) have been applied in (Feng et al., 2003) and proved to give satisfying results. As the collection is mostly focused on popular and classical music, the "fear" adjective has been extended to a larger category called "mysterious." Using Wordnet[9] we have joined the possible playlists together in the following four categories : happy, sad, angry and mysterious. After that, a musicologist validated each song label.

We obtained a ground truth database of 191 songs with the distribution in mood shown in Table 3.3. For each song, there is only one mood label. It is not an equal distribution but there is enough data in each category to experiment with the CB similarity.

Table 3.3: Mood distribution of the ground truth.

Mood	Songs
Happy	67
Sad	61
Angry	34
Mysterious	29

[8] http://www.magnatune.com/moods/

[9] http://wordnet.princeton.edu/

3.3.2. Experimental results

The goal of this experiment is to prove empirically how content-based similarity can help to propose labels to yet unlabeled songs, and thus reducing the hard effort of manually annotating songs.

We present two different experiments. The first one propagates labels that are related with the style of the piece, whereas the second experiment deals with mood labels. The problem with the Magnatune collection is that there is only one human that annotated the tracks, when normally a ground truth of this nature should be pair–reviewed. Yet, we validated a large amount of the annotated songs by listening to them.

Propagation of music style labels

The ground truth for the style experiment consists of 29 different labels (like *Rock*, *Instrumental*, *Classical*, *Relaxing*, etc.), and 5481 annotated songs.

The evaluation process was the following, for a partially annotated collection (10%,..., 50%), we use the CB module to get the *ith*–similar (i=10, 20 and 30) songs —and their tags— to a given one, to propose tags based on the tags from these similar songs. However, we did not propose those tags that appeared with a frequency less than 20% (i.e., a `voting threshold` of 0.2)

Evaluation metrics. The metrics used to evaluate the styles experiments were initially Precision/Recall and F_2-Measure (giving more weight to Recall). In our case, Recall seems to be more informative since our purpose is to know how well the tags can be propagated. We also used Spearman's rank correlation coefficient (or Spearman ρ) to take into account the frequencies (i.e. ranking) of the tags obtained from the similar songs.

Results. For the style experiment, we ran different configurations and we computed the average metrics. A special case is when using the 100% annotated songs (see the results in Table 3.4). This experiment is used to test whether the CB similarity is good for propagating labels. There are four different configurations when retrieving the most similar songs to a given one: do not apply any constraint, or filter by artist/album. The constraints, then, are: filtering the similarity results by same Artist, same Album, or by same Artist and Album. The latter case makes only sense when the songs appears in compilations, various artists albums, etc. When filtering by artist or by album we make sure that the most similar songs to a given one are not from the same artist or the same album. That of course decreases the Precision/Recall measure. We can see from the results, that to achieve more precision and recall when applying a constraint, we need to increase the number of similar songs, which makes sense because we are not taking into account similar songs that are closer to a given one.

Table 3.4: Experiments with the 100% annotated collection. The Precision/Recall measure, the F_2-measure and the Spearman ρ measure are proportional to the number of similar songs. When constraints are present, these measures decrease.

Sims.	Constraint	P	R	F_2	ρ
10	None	0.56	0.84	0.72	0.51
	Artist	0.41	0.58	0.51	0.23
	Album	0.50	0.71	0.62	0.34
	Artist & Album	0.43	0.59	0.53	0.19
20	None	0.56	0.82	0.71	0.49
	Artist	0.48	0.61	0.56	0.26
	Album	0.53	0.72	0.64	0.35
	Artist & Album	0.48	0.61	0.56	0.24
30	None	0.60	0.77	0.70	0.45
	Artist	0.50	0.58	0.55	0.28
	Album	0.56	0.67	0.63	0.37
	Artist & Album	0.50	0.59	0.55	0.27

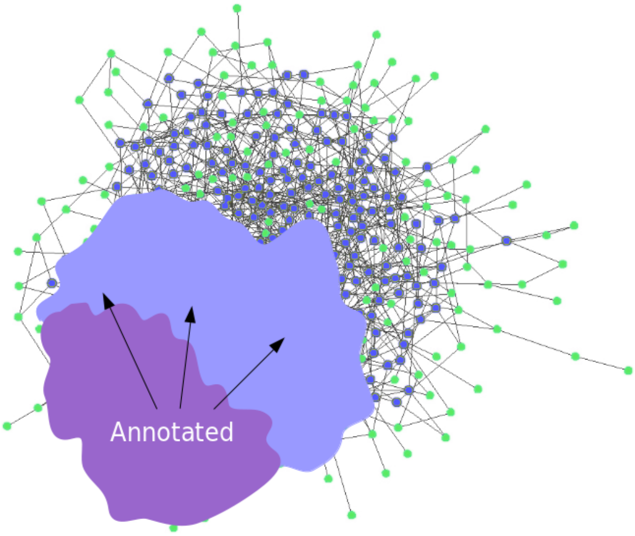

Figure 3.2: Propagating tags in a partially annotated collection.

Now, Table 3.5 shows the results of propagating a partially annotated collection (see Figure 3.2). The Spearman ρ coefficient, as well as Precision/Recall and F_2-measure, grows when increasing the percentage of songs annotated in the collection. Interestingly enough, the values decrease when increasing the number of neighbors (from 10 to 30) for a given song.

Finally, we propose another experiment that is to automatically annotate songs

Table 3.5: Experiments with the 20%, 40% and 50% annotated collection. The Precision, Recall and F_2-measure and the Spearman ρ values grow with a higher percentage of annotated songs, and a smaller number of similar songs.

Annotation	Sims.	P	R	F_2	ρ
20%	10	0.32	0.29	0.30	0.24
	20	0.22	0.17	0.19	0.16
	30	0.08	0.05	0.06	0.06
40%	10	0.57	0.59	0.58	0.43
	20	0.56	0.52	0.53	0.41
	30	0.49	0.39	0.42	0.34
50%	10	0.61	0.67	0.64	0.47
	20	0.61	0.61	0.61	0.45
	30	0.57	0.51	0.53	0.41

in a music collection by means of the propagation process. The results are presented in Table 3.6. It is clear that the percentage of songs automatically annotated by CB similarity increases when the number of already annotated songs grows. But, we can see an interesting exception here, that is the 40% annotated collection performs better (up to 38.68% new propagated labels, with a low Recall 0.4) than the 50% one. This could be due to the random process of splitting the ground truth and the test set from the collection. Furthermore, we can see how the percentage of songs automatically annotated is inversely proportional to the number of similar songs used by the CB similarity module (in contrast with the results from the 100% annotated collection, see Table 3.4, when applying any constraint).

Propagation of mood labels

Evaluation metrics. To evaluate the mood results, we used two measures. First we wanted to check if the system was able to guess the correct mood label (there is only one possible label per song). We evaluated the Precision just considering the first result using Precision at 1, also called P@1.

$$P@1 = \begin{cases} 1, & \textit{best proposed label} = \textit{real label} \\ 0, & \textit{otherwise} \end{cases} \tag{3.6}$$

We averaged this value over all the examples. This metric helps us to understand if the system can predict the correct mood label. However it does not take into account the relative frequencies. Then another measure would be needed to evaluate this aspect. We weighted the frequencies of the proposed label and normalized to compute a weighted Precision at 1, that we will call

Table 3.6: Extending annotations of a music collection by means of CB similarity. We observe that the propagation grows with a smaller number of similar songs and a higher percentage of annotated songs, except for the case of 40% and 50%.

		Propagation with Recall		
Annotation	Sims.	> 0.8	> 0.6	> 0.4
20%	10	17.515%	21.365%	24.977%
	20	8.666%	12.352%	15.453%
	30	2.554%	3.758%	5.145%
40%	10	28.01%	33.46%	38.68%
	20	22.50%	28.92%	34.32%
	30	15.22%	20.82%	26.22%
50%	10	26.77%	31.62%	35.92%
	20	22.66%	28.74%	33.37%
	30	17.48%	23.15%	28.44%

Table 3.7: Confusion matrix for the mood experiment with a 100% annotated collection.

GT \Predicted	Angry	Happy	Mysterious	Sad
Angry	27	7	1	1
Happy	4	55	1	2
Mysterious	8	6	7	5
Sad	4	16	2	35

wP@1. It is equal to the frequency value of the correct label over the sum of all the proposed label frequencies:

$$wP@1 = \frac{freq.\,correct\,label}{\sum freq.\,proposed\,labels} \tag{3.7}$$

Results. To have an overview of the system performance for each mood, we built a confusion matrix in Table 3.7. It has been computed using 100% of the collection annotated. Each row gives the predicted mood distribution (considering only the best label) for each mood in the ground truth. Looking at the confusion matrix we observe that a CB similarity approach can propagate relatively well the "happy," "angry," and "sad" labels. However the "mysterious" label does not give good results. We can explain this by the fact that it might be the most ambiguous concept of these categories. Table 3.8 presents the average P@1 and wP@1 values per mood.

Table 3.8: P@1 and wP@1 values averaged for each mood.

	Angry	Happy	Mysterious	Sad	All
P@1	0.72	0.89	0.27	0.61	0.62
wP@1	0.65	0.62	0.22	0.59	0.52

It confirms what we have in the confusion matrix, the “happy” category gives the best result. However looking at the values of wP@1, we note that if “happy” is the most guessed mood, the system gives more reliability to its results about the label “angry.”

In our last experiment we wanted to evaluate how well the mood labels can be propagated if we annotate just partially the collection. We computed the P@1 for 70%, 50% and 30% of the database and obtain the results written in Table 7. It shows that for 30% of the collection annotated, the system can propagate correctly the tags up to 65% of the collection.

Table 3.9: Evaluation of the mood label propagation with the initially percentage of annotated songs.

Initial annotation	P@1	Correctly annotated after prop.
70%	0.60	88%
50%	0.44	72%
30%	0.50	65%

As the content–based approach may not consider important aspects that can infer the mood, all these performances should be improved by using dedicated descriptors and meta-data, like information about the title, the style or the lyrics.

3.3.3. Discussion

Our objective was to test how the content–based similarity can propagate labels. For styles, we have shown that with a 40% annotated collection, we can reach a 78% (40%+38%) annotated collection with a recall greater than 0.4, only using content–based similarity. In the case of moods, with a 30% annotated collection we can automatically propagate up to 65% (30% +35%). These results are quite encouraging as content–based similarity can propagate styles and moods in a surprisingly effective manner. Of course there are some limitations as the example of the “mysterious” label, the concept has to be clearly encoded in the music for the content–based propagation to work. For the moods we will try to experiment with a larger database, different taxonomies and more concepts. With our current mood results it may not be possible to generalize but it shows the potential of the technique. In general, to enhance the performance of such an automatic annotation system we would use a hybrid approach combining content–based, user feedback and social networks informations. But as shown by the satisfying results, our propagation system based on content–based similarity would already ease a lot the annotation process of huge music collections.

3.4. Experiment 2: Freesound.org dataset

3.4.1. Introduction

Freesound.org is a collaborative sound database where people from different disciplines share recorded sounds and samples under the Creative Commons license, since 2005. The initial goal was to giving support to sound researchers, who often have trouble finding large sound databases to test their algorithms. After four years since its inception, *Freesound.org* serves more than 23,000 unique visits per day. Also, there is an engaged community—with almost a million registered users—accessing more than 66,000 uploaded sounds.

Yet, only few dozens of users uploaded hundreds of sounds, whilst the rest uploaded just a few. In fact, 80% of the users uploaded less than 20 sounds, and only 8 users uploaded more than one thousand sounds each. It is worth noting that these few users can highly influence the overall sound annotation process.

Tag behavior

In this section we provide some insights about the tag behavior and user activity in the *Freesound.org* community. We are interested in analyzing how users tag sounds assets, as well as the concepts used when tagging. The data, collected during March 2009, consists of around 66,000 sounds annotated with 18,500 different tags

Figure 3.3 shows the number of tags used to annotate the audio samples. The x-axis represent the number of tags used per sound. We can see that most of the sounds are annotated using 3–5 tags. Also, around 7,500 sounds are insufficiently annotated using only 1 or 2 tags. These sounds represent more than 10% of the whole collection. It would be desirable, then, to—automatically—recommend relevant tags to these scarcely annotated sounds, enhancing their descriptions. This is the main goal of the experiments presented in Section 3.4.3.

Interestingly enough, in (Cano, 2007), the author analyzed a sound effects database, which was annotated by only one expert. A similar histogram distribution to the one presented in Figure 3.3 was obtained. Specifically, most of the sounds were annotated by the expert using 4 or 5 tags, as it is our case. This could be due to human memory constraints when assigning words

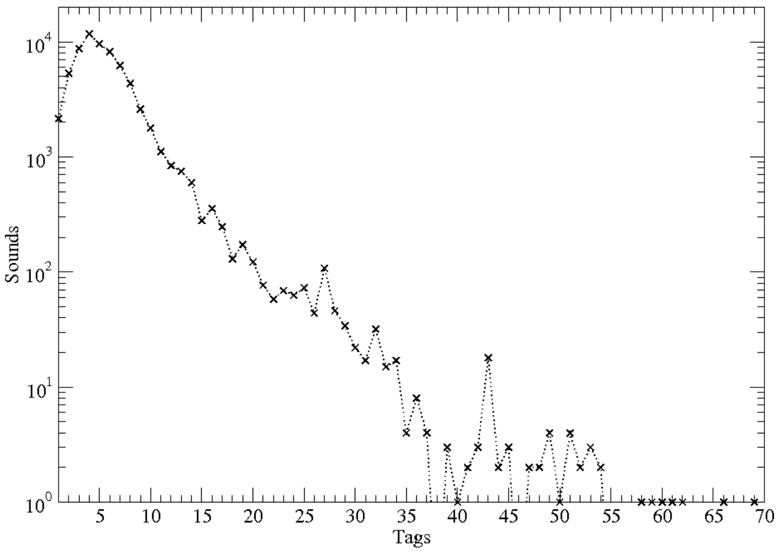

Figure 3.3: A linear–log plot depicting the number of tags per sound. Most of the sounds are annotated using 3–5 tags, and only a few sounds are annotated with more than 40 tags.

to sounds or to any object, in order to describe them (Miller, 1956). Based on Figure 3.3, we classify the sounds in three different categories, according to the number of tags used. Table 3.10 shows the data for each class.

Table 3.10: Sound–tag classes and the number of sounds in each category.

	Tags per sound	**Sounds**
Class I	1–2	7,481
Class II	3–8	42,757
Class III	> 8	7,148

Tag frequency distribution is presented in Figure 3.4. The x-axis refers to the 18,500 tags used, ranked by descending frequency. On the one hand, 44% of the tags were applied only once. This reflects the subjectivity of the tag process. Thus, retrieving these sounds in the heavy tail area is nearly impossible using only tag–based search (to overcome this problem, *Freesound.org* offers a content–based audio similarity search to retrieve similar sound samples). On the other hand, just 27 tags were used to annotate almost the 70% of the whole collection. The best fit of the tag distribution is obtained with a log–normal function, $\frac{1}{x}e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$, with parameters mean of log $\mu = 1.15$, and standard deviation of log, $\sigma = 1.46$ (Clauset et al., 2007).
The top–5 most frequent tags are presented in Table 3.11, and it gives an idea

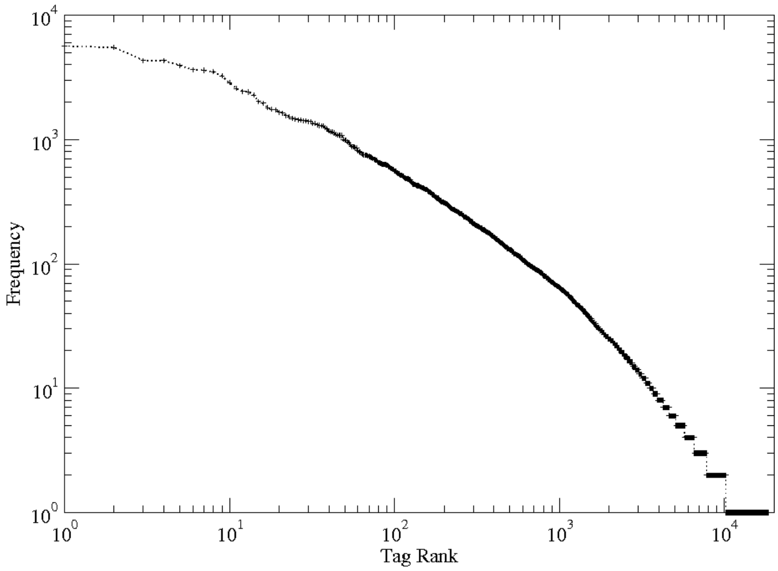

Figure 3.4: A log–log plot showing the tag distribution in *Freesound.org*. The curve follows a log–normal distribution, with mean of log $\mu = 1.15$, and standard deviation of log, $\sigma = 1.46$.

about the nature of the sounds available in the *Freesound.org* collection. Field–recording is the most frequent tag used to describe 6,787 different sounds. All these frequent tags are very informative when describing the sounds, in contrast to the photo domain in *flickr.com*, were popular tags are considered too generic to be "useful" (Sigurbjörnsson & van Zwol, 2008).

Table 3.11: Top–5 most frequent tags from Figure 3.4.

Rank	Tag	Frequency
1	field–recording	6,787
2	noise	5,650
3	loop	5,487
4	electronic	4,329
5	synth	4,307

Tag categorization

In order to understand the vocabulary that the *Freesound.org* community uses when tagging sounds, we mapped the 18,500 different tags to broad categories (hypernyms) in the Wordnet[10] semantic lexicon. In some cases, a given tag

[10] http://wordnet.princeton.edu/

matches multiple entries, so we bound the tag (noun or verb) to the highest ranked category. The selected Wordnet categories are: *(i)* artifact or object, *(ii)* organism, being, *(iii)* action or event, *(iv)* location, and *(v)* attribute or relation. Yet, 20.3% of the tags remain unclassified.

Most of the tags (38%) are related with objects (e.g., *seatbelt*, *printer*, *missile*, *guitar*, *snare*, etc.), or about the qualities and attributes of the objects (30%); such as state attributes (*analog*, *glitch*, *scratch*), or magnitude relation characteristics (*bpm*). Then, some tags (19%) are classified as an action (*hiss*, *laugh*, *glissando*, *scream*, etc.), whilst 11% are related with organisms (*cat*, *brass band*, etc.). Finally, only a few tags (2%) were bound to locations (e.g., *iraq*, *vietnam*, *us*, *san francisco*, *avenue*, *pub*, etc.). Therefore, we conclude that the tags are mostly used to describe the objects that produce the sound, and the characteristics of the sound. In this case, the wisdom of crowds concords with the studies of (Schaeffer, 1966) and (Gaver, 1993). The former study focused on the attributes of the sound itself without referencing the source causing it (e.g *pitchiness*, *brightness*), while the latter introduced a taxonomy of sounds, on the assertion that they are produced by means of interaction of materials.

3.4.2. Dataset

The sounds selected for the experiments were a subset of the Class I (see Table 3.10). We selected those sounds whose tags' frequency was very low (i.e. rare tags, in the ranking of $\sim 10^4$ in Figure 3.4). In fact, all the sounds which were annotated with one tag whose frequency was equal to 1 were selected. Also, for the sounds annotated with 2 tags, we selected those which had at least one tag with frequency 1. The test dataset for the experiments consists of 260 sounds. The goal here is to automatically extend the annotation of these sounds, insufficiently annotated with one or two very rare tags. Table 3.12 presents additional information about this test dataset.

Table 3.12: Additional information about the Freesound.org test dataset, gathered from `www.freesound.org` during March, 2009.

#Tracks	#Tags	Tags/Track	Top Tags	Bottom Tags
260	399	$\mu = 1.76$	Drum (17)	Chords (1)
		$\sigma = 0.43$	Heavy (5)	Rain2 (1)
			Synth (5)	Wash-hand (1)
			Wind (4)	Time (1)
			Water (3)	Clapsticks (1)

3.4.3. Experimental results

Our goal is to evaluate the quality of the recommended tags, for some specific sounds available in *Freesound.org.* By means of content–based audio similarity, our algorithm selects a set of candidate tags for a given sound (autotagging process). Then, the evaluation process is based on human assessment. Three subjects validated each candidate tag for all the sounds in the test dataset.

We used a nearest neighbor classifier (k–NN, $k = 10$) to select the tags from the most similar sounds of a given sound. The choice of a memory–based nearest neighbor classifier avoids the design and training of every possible tag. Another advantage of using an NN classifier is that it does not need to be redesigned nor trained whenever a new class of sounds is added to the system. The NN classifier needs a database of labeled instances and a similarity distance to compare them. An unknown sample will borrow the metadata associated with the most similar registered sample.

Procedure

Our technique for calculating the candidate tags consists on finding the *10–th* most similar sounds from the *Freesound.org* database, for a given seed sound of the test dataset. That is, given a seed sound, we get the tags from the similar sounds. A tag is proposed as a candidate if it appears among the neighbors over a specific threshold. For example, a threshold of 0.3, means that a tag is selected as candidate when it appears at least in 3 sounds of the 10 nearest neighbors. This way we select the set of candidate tags for each sound in the test dataset.

The experiments have been computed using two thresholds: 0.3 and 0.4. When using a threshold of 0.3 the number of candidate tags is higher than for 0.4, but also there are more “noisy” or potentially irrelevant tags, since it is using a less constrained approach. Afterwards, all the candidate tags will be evaluated by human assessment. The differences between both thresholds is presented in Section 3.4.3.

Evaluation

In order to validate the candidate tags for the test sounds, we use human assessment. The aim is to evaluate the perceived quality of the candidate tags. It is worth noting that neither Precision nor Recall measures are applicable as the test sound contains only two or less tags, and these are very rare in the vocabulary. We performed a listening experiment where the subjects were asked to listen to the sounds, and decide whether they agreed or not with the candidate tags. For each candidate tag, they had to select one of these options: *Agree* (recommend candidate tag), *Disagree* (do not recommend), or *Don’t know.* Each sound was rated by three different subjects.

Similar to (Turnbull et al., 2007a), to evaluate the results we group human responses for each sound s, and score them in order to compact them into a single vector per sound. The length of the vector is the number of candidate tags of s. Each value of the vector, w_{s,t_i}, contains the weight of the subjects' scores for a candidate tag t_i in sound s. If a subject agrees with the candidate tag, the score is $+1$, -1 if disagrees, and 0 if she does not know. The formula for calculating the weight of the candidate tag in s is:

$$\mathbf{w}_{s,t_i} = \frac{\#(PositiveVotes) - \#(NegativeVotes)}{\#Subjects} \tag{3.8}$$

A candidate tag is recommended to the original sound if $\mathbf{w}_{s,t_i}$ is greater than zero, otherwise, the tag is rejected (either because it is a bad recommendation, or the subjects cannot judge the quality of the tag). For example, given a candidate tag t_i for s, if the three subjects scored, respectively, $+1$, -1, $+1$ (two of them agree, and one disagree), the final weight is $\mathbf{w}_{s,t_i} = 1/3$. Since this value is greater than zero, t_i is considered a good tag to be recommended. Furthermore, we use $\mathbf{w}_{s,t_i}$ to compute the confidence agreement among the subjects. First, we consider all the sounds where the system proposed j candidate tags, S_j. We sum, for each sound $s \in S_j$, the weights of all the candidate tags t_i whose values were greater than zero. Then, we divide this value with the total score that the candidate tags would had if all the subjects would agree. The formula for calculating the agreement of S_j sounds, A_j, is:

$$A_j = \frac{\sum_{s \in S_j} \left[\mathbf{w}_{s,t_i} > 0\right]}{\#Subjects \cdot \left[\sum_{s \in S_j} length\,(s)\right]} \tag{3.9}$$

Similarly, to compute the agreement of the bad candidate tags, we use the weights of candidate tags whose values were lesser than zero ($\mathbf{w}_{s,t_i} < 0$), in the numerator of the Equation 3.9. Finally, to get the total agreement for all the sounds in the test set, A_{total}, we use the weighted mean of all A_j, according to the number of sounds in A_j.

Results

Perceived quality of the recommended tags. Using 10–NN and the content–based audio similarity, and setting a threshold of 0.3, the system proposed a total of 781 candidate tags, distributed among the 260 sounds of the test dataset. Besides that, setting a threshold of 0.4 the system proposes 358 candidate tags, which represents almost the half compared with a threshold of 0.3.

Table 3.13 shows the human assessment results. As expected, a slightly higher percentage of candidate tags were recommended with a threshold of 0.4 (66.23%). Yet, using a threshold of 0.3, more than half of the candidate tags (56.6%) were finally recommended to the original sounds, with an agreement confidence of

Table 3.13: Percentage of recommended tags, with confidence agreement among the subjects. The table shows the results using thresholds 0.3 and 0.4 (in parenthesis, it is shown the total number of candidate tags).

Threshold	**Recommend tag**	**%**	$A_{\textbf{total}}$
0.3 (781)	Yes	56.60%	0.74
	No	31.59%	0.62
	Don't know	11.41%	—
0.4 (358)	Yes	66.23%	0.78
	No	23.11%	0.58
	Don't know	10.66%	—

0.74. This human agreement is sufficiently high to rely on the perceived quality of the recommended tags. The rest of the candidate tags (43.4%) were not recommended, either because the tags recommended were not appropriated (31.59%), or the tags were not sufficiently informative (11.41%). Even though with a threshold of 0.3 we get less percentage of recommended tags, the absolute number of candidate tags is more than twice the ones with a threshold of 0.4. Therefore, we can consider a threshold of 0.3 a good choice for this task.

Recommended tags per class. On the one hand, using a threshold of 0.4 we are able to enhance the annotation of half of the sounds (128 sounds out of 260). On the other hand, with a threshold of 0.3, we have enhanced the annotation of 200 sounds, which represent the 77% of the sounds in the test dataset used. The rest of the sounds (60) from the test set did not get any plausible tags to extend its current annotation.

Table 3.14 shows the results using a threshold of 0.3, and it classifies the 200 autotagged sounds according to the classes defined in Table 3.10. Originally, all the test sounds belonged to Class I. We can observe now the number of sounds per class, after extending the annotation of these 200 sounds. Note that most of the sounds have 3 or more tags (Class II), and some even have more than 8 tags (Class III). However, there are 20 sounds still belonging to Class I. This happens because before the experiment they only had one tag, and now they have another one, the one recommended.

The results obtained so far look promising; using a simple classifier we were able to automatically extend sound annotations that were difficult to retrieve. Furthermore, due to the classifier method used (k–NN), there is a strong correlation among the more frequently proposed tags, and their frequency of usage (rank position in Figure 3.4). The ten most proposed tags are also in the top–15 ranking of frequency use. Although our approach is prone to popular tags, once the sounds are autotagged it allows the users to get a higher recall of those scarcely annotated sounds when doing a keyword–based search.

Table 3.14: Number of sounds in each category, after automatically extending the annotations of 200 sounds from the test dataset.

	Tags per sound	**Sounds**
Class I	1–2	20
Class II	3–8	171
Class III	> 8	9

3.4.4. Conclusions

In this section we presented an analysis of the *Freesound.org* collaborative database, where the users share and browse sounds by means of tags, and content–based audio similarity search. First we studied how users annotate the sounds in the database, and detected some well–known problems in collaborative tagging, such as polysemy, synonymy, and the scarcity of the existing annotations.

Regarding the experiments, we selected a subset of the sounds that are rarely tagged, and proposed a content–based audio similarity to automatically extend these annotations (autotagging). Since the sounds in the test set contained only one or two rare tags, neither precision nor recall were applicable, so we used human assessment to evaluate the results. The reported results show that 77% of the test collection were enhanced using the recommended tags, with a high agreement among the subjects.

3.5. Experiment 3: Statistical testing on the CAL500 dataset

The purpose of this third experiment is two fold. First, we want to investigate how different parameters (as defined in Section 3.2.4) can affect the performance of our algorithm. Moreover, we analyze if the impact of the different parameters is statistically significant. Second, after selecting a parameter configuration from the first step, we evaluate our algorithm against a considerable number of state of the art approaches.

3.5.1. Dataset

The CAL500 dataset (Turnbull et al., 2007a), is a music collection consisting of 500 songs from 500 artists, with a vocabulary of 174 tags, grouped in 6 different categories. The categories are emotion, musical genre, instrumentation, solo instrument, usage and vocal characteristics. Table 3.15 shows some additional information about this dataset.

The collection is distributed with the basic metadata (artists and tracks names), the annotation matrix, and a set of 10,000 39–dimensional feature vectors

Table 3.15: Additional information about the CAL500 Dataset.

#Tracks	#Tags	Cat./Track	Tags/Track	Top Tags	Bottom Tags
500	174	$\mu = 5.85$ $\sigma = 1.13$	$\mu = 26.04$ $\sigma = 5.74$	Recorded (438) Male Lead Vocals (335) Texture Electric (324) Not Angry / Aggressive (314) Not Bizarre / Weird (291)	Monotone (6) Swing (5) Bebop (5) Soul (4) With the family (4)

per music piece, consisting in 13 MFCC coefficients, 13–Delta MFCC coefficients and 13 Delta–Delta MFCC coefficients. The MFFC coefficients are extracted from the audio file with a sliding, half–overlapping short–time window ($\tilde{1}2$ msec), together with their corresponding deltas, making a total of about 10,000 39–dimensional vectors per minute of audio. Nevertheless, Turnbull et al. (2007a) found that randomly sub-sampling the delta cepstrum feature vectors, resulting in 10,000 feature vector representation per audio file, reduces the computation time for parameter estimation without sacrificing much overall performance. The audio collection, on the other hand, is not distributed, due to copyright reasons. Thus, we matched the CAL500 dataset against our own music collection.

3.5.2. Parameter selection

The aim of this experiment is to investigate how different parameters can affect the performance of our algorithm, following the idea of Aucouturier & Pachet (2004) for music tag classification. We create different instances of our algorithm, by tuning the parameters mentioned in Section 3.2.4. A 10–fold cross-validation is performed to evaluate the algorithm in both annotation and retrieval tasks (please refer to Section 2.4.1 for more details regarding the evaluation measures). We compare the mean `F-measure` for annotation, and `Mean Average Precision` for retrieval. We also check whether the impact of any of the different parameters are statistically significant (Flexer, 2006).

Table 3.16 presents the different values we use in our experiment. The variables to study are the amount of variance that the dimension-reduced feature representation covers from the original data, the use of high level features, the distance metric, and the number of nearest neighbors (the k in k–NN). To reduce the number of combinations, we sample the range of nearest neighbors to the subset $(1, 2, 5, 10, 15, 18)$. This gives $(5 \times 2 \times 2 \times 6) + (5 \times 1 \times 1 \times 6) = 150$ different combinations. It should be noted that choosing to use high level features does not affect the Kullback-Leibler distance, since it is computed over

Table 3.16: List of the different parameter values used for tuning our autotagging algorithm.

Parameter	Values
PCA covered variance	75%, 80%, 85%, 90%, 100%
High level features	Yes, No
Distance metric	Euclidean, Kullback-Leibler, Linear Combination
Number of nearest neighbors	$(1, 2, 3, ..., 20)$

the MFCC covariance matrix only.

Annotation

The hypothesis of running different instances of the algorithm by tuning the 4 available parameters[11] is manifold.

1. We want to check if reducing the covered variance of the original feature representation with Principal Component Analysis from 100% to 75% — which decreases the dimension significantly — affects the performance of the algorithm.

2. Different results for F–measure should be achieved by using a Euclidean distance over the reduced space, or the Kullback-Leibler over the MFCC covariance matrix only, or a Linear Combination of both, since we are changing both the data representation and the way we measure the similarity distance.

3. High level features, which are probability estimations of high level concepts such as `genres` or `moods`, when used, should have a positive effect on the evaluation results. First, these features are capturing, by definition, more human readable concepts. For the same reason, it is more likely that tag correlation might be found with these features while learning. For example, if a song has, a priori, a high probability of having genre *classical* as an audio high level feature, then it is unlikely that this song would be annotated with instrument *electric guitar*

4. Different number of similar songs, that is, different values for k, should lead to a higher (or lower) precision/recall values, thus affecting the F–measure [12].

[11] As mentioned earlier in Section 3.2.4 the voting threshold parameter is set to 0.2 from empirical evaluations.

[12] It is worth recalling that F–measure is a weighted harmonic mean, and not an arithmetic mean. Hence it will depend on the goodness of both precision and recall.

We chose the mean per–song *F–measure* — averaged over a 10–fold cross–validation basis — as a measure to evaluate the performance of the different classifiers. Since the evaluation results are specific for each song, we considered the song as a factor too, and thus we tested and validated our hypothesis by performing a multi-factor within–subjects ANOVA, also known as repeated measures ANOVA (Hinkelmann et al., 2005).

The results indicate that there was no statistically significant difference among the different values for the PCA covered variance parameter ($p = 0.172$ for Mauchly's spherificity and $p = 0.207$ for lower-bound). This means that there is no statistical evidence to support the hypothesis that reducing the amount of covered variance would significantly affect the performance. Nevertheless, since there was no significant change in F–measure while decreasing the amount of covered variance, we choose the parameter value of 75%, which compresses the feature vector dimensions considerably ($\sim$ 29-30 dimensions from the original 611), and thus reducing the complexity of the algorithm. Interestingly, Seyerlehner et al. (2010) show, via experimental results, that a feature set capturing about 70%–80% of the total variance achieves optimal results in terms of tag classification. On the other hand, significant differences were found ($p < 0.001$) with the rest of the parameters, namely high level features, distance metrics and number of nearest neighbors (k). Interestingly, the multi-factor ANOVA allows us to consider the combination of factors as well. In this case, the combination of high level features and distance metrics was also significant ($p < 0.001$). Tukey's test is then applied as a post–hoc test to evaluate the differences among the different values for each parameter. Figure 3.5 depicts the box–and–whisker plot for the parameters. We found that using high level features combined with Euclidean distance increases the F–measure of the autotagging algorithm. This combination outperforms the results using the Kullback-Leibler divergence over the MFCC covariance matrix, which can add more support to the assertion that MFCC coefficients are not a sufficient representation of music signals for tasks such as music tag classification (Aucouturier & Pachet, 2004).

Surprisingly, it is found that by using 18 nearest neighbors, which is a relatively high number, the global (per–song) performance of the algorithm improves significantly. This may be due to multiple reasons. First, it depends on the distribution of the audio feature vectors in the d-dimensional space, the tag frequency, and the number of tags per song. Second, having 18 nearest neighbors will make the voting threshold of 0.2 to discard tags that are used in less than $0.2 \times 18 = 3.6 \approx 4$ songs. Hence, although increasing the probability of having more tags to choose from (recall), the voting threshold also increases, which may have a positive effect on the precision.

It is worth mentioning that a value of $k = 18$ is not aligned with the results previously achieved in Sections 3.3 and 3.4, where the best results were obtained with $k = 10$ (in this dataset, the marginal f–measure mean difference between $k = 18$ and $k = 10$ is 0.012). These findings might suggest that the

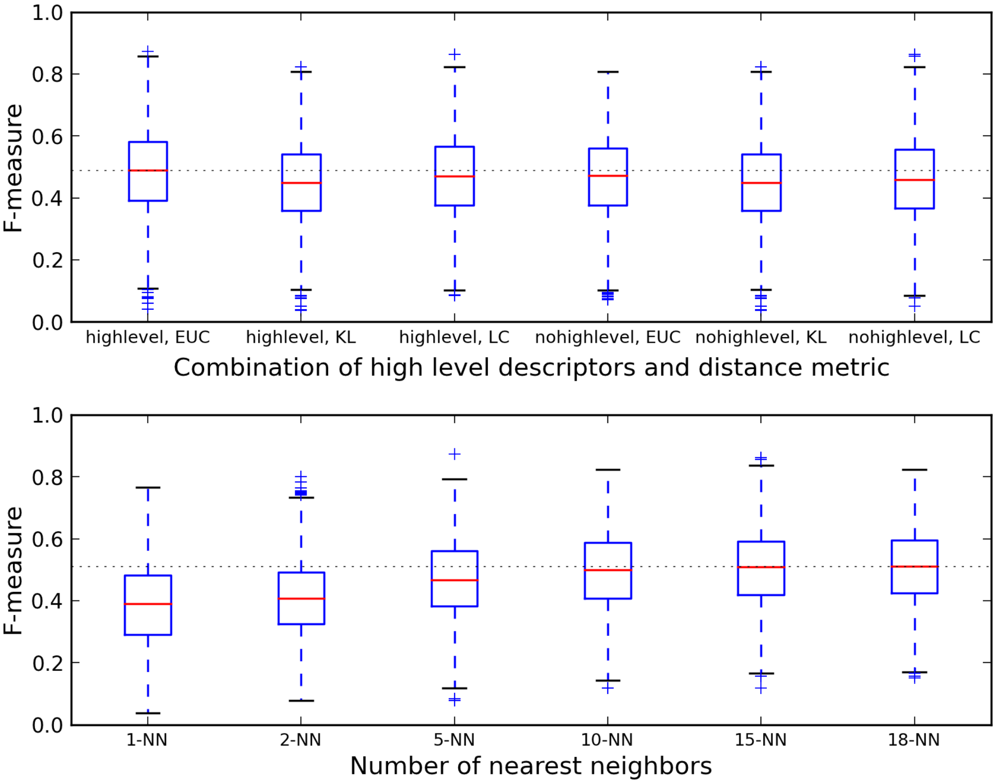

Figure 3.5: Box and whisker plot of the parameter selection for the music annotation task, using the CAL500 dataset.

results achieved with this last experiment are biased to the CAL500 collection. Nevertheless, we will use the parameter configuration $k = 18$ for further experiments, where different music collections will be used to assess its validity.
Last but not least, our decision to use within–subject tests instead of between–subject test was also supported by Figure 3.5, where it can be observed the large variability of the subjects (in this case, songs).

Retrieval

In the music retrieval task, we are interested in evaluating how well our algorithm returns a relevant list of songs, given a query tag. As in the case of music annotation, there are 4 parameters that can be tuned to modify the performance of the system. The parameters values are chosen as described in Table 3.16.
To compare the performance of the different configurations, Mean Average Precision (MAP) is used. The voting threshold parameter is disabled in this scenario, since we are ranking all the tags in the vocabulary (see Section 3.2.3

for more details).

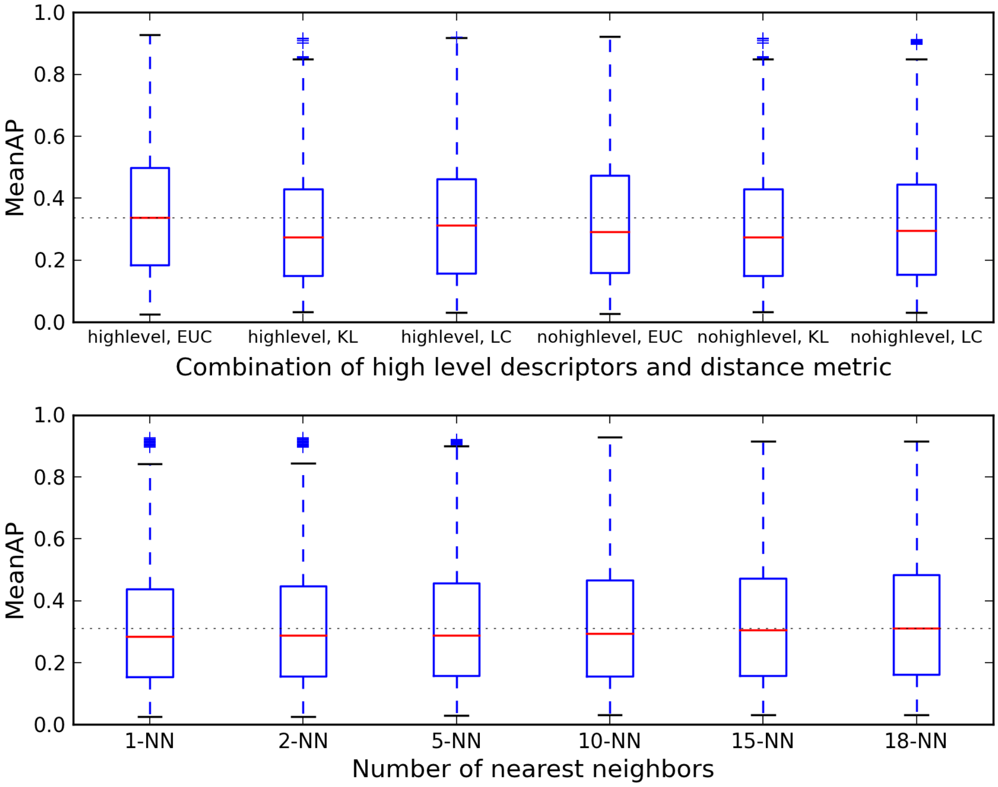

Figure 3.6: Box and whisker plot of the parameter selection for the music retrieval task, using the CAL500 dataset.

After running the different configurations and performing a multi-factor within–subjects ANOVA, we obtained the same results as in the case of music annotation. That is, changing the PCA covered variance did not significantly affect the performance ($p = 0.244$ for Mauchly's sphericity and for lower-bound), whilst the other 3 parameters presented a statistical significance ($p < 0.001$). Again, the combination of high level features and distance metrics was also significant ($p < 0.001$). The difference is even bigger in the high level and metric parameters. As for the number of nearest neighbors, while still agreeing with the results found in the music annotation evaluation —that is to say, a higher number of similar songs produces statistically better results—, the difference is now lower. This can be due to the fact of disabling the voting threshold parameter, which had a positive effect in the annotation precision.
Based on the statistical test results in both annotation and retrieval tasks, we select the parameter values that give the best results. Henceforth, if it is not explicitly defined, our parameter configuration for the following experiments is as presented in Table 3.17.

Table 3.17: Final parameter values, after performing statistical tests to check which is the best configuration for our algorithm.

Parameter	Value
PCA covered variance	75%
Use high level audio features	Yes
Distance metric	Euclidean
Number of nearest neighbors	18–NN

3.5.3. Experimental results

In this section, the autotagging algorithm for music annotation and retrieval is evaluated, using the CAL500 dataset. Additionally, the results are compared with other state of the art autotagging algorithms which used the same dataset for evaluation (Bertin-Mahieux et al., 2008; Coviello et al., 2011; Hoffman et al., 2009; Turnbull et al., 2008b).

The audio features extracted are illustrated in Table 3.1. Then, a 10-fold cross validation is performed over the whole dataset. That is, each fold iteration consists 450 songs for training and 50 songs for testing.

Annotation

In this experiment we focus on the annotation task. That is, we want to test the ability of the autotagging algorithm to annotate songs with few relevant words. Table 3.18 presents quantitative results for music annotation. The results are means and standard errors computed from 10–fold cross validation. Standard IR measures `Precision`, `Recall` and `F-measure` are used for evaluation. We annotate songs with 10 and 26 tags. 26 corresponds to the average annotation in the Ground Truth dataset (see Table 3.15), while 10 is taken from Turnbull et al. (2008b), in order to select which algorithm instances will be used for comparison.

The "Random" model is taken from Turnbull et al. (2008b). This model samples tags, without replacement, from a multinomial distribution of the prior probability of tags, which is estimated using the tag's frequency. Whilst the standard IR measures such as mean per–song Precision, Recall and F–measure indicate how well the systems proposes tags to new, unseen songs, tag coverage informs us whether the autotagging algorithm is able to predict all the tags in the vocabulary.

It is interesting to note that the system having the best overall per–song F–measure, 18–NN, has the poorest tag coverage. In order to tackle this issue, results were also generated for a 2–NN classifier. Our rationale is that disabling the voting threshold and reducing the number of nearest neighbors should increase the number of predicted tags, without affecting the global (per–song)

Table 3.18: Music annotation results for the CAL500 dataset, computed from 10–fold cross validation. A = Annotation length. Tag coverage measures how many tags, from the total 174 of the dataset vocabulary, the system is able to predict. We test our k-NN algorithm with the configuration given in Table 3.17, which uses $k = 18$, and also with $k = 2$. CBDC refers to the class–based distance classifier mentioned in Section 3.2.3. EUC = Euclidean, COS = cosine, and MAHAL = Mahalanobis distance.

Algorithm	A	Tag coverage	Precision		Recall		F-Measure
Random	10	174	0.144	(0.005)	0.059	(0.002)	0.079
	26	174	0.149	(0.003)	0.149	(0.003)	0.149
18–NN (EUC)	10	66	**0.690**	(0.008)	**0.271**	(0.002)	**0.390**
	26	103	**0.529**	(0.006)	**0.533**	(0.005)	**0.531**
2–NN (EUC)	10	122	0.551	(0.007)	0.215	(0.003)	0.309
	26	170	0.432	(0.006)	0.433	(0.005)	0.433
CBDC (EUC)	10	**174**	0.287	(0.008)	0.109	(0.003)	0.158
	26	**174**	0.301	(0.006)	0.296	(0.005)	0.298
CBDC (COS)	10	173	0.275	(0.008)	0.104	(0.003)	0.151
	26	**174**	0.298	(0.006)	0.294	(0.005)	0.296
CBDC (W.EUC)	10	163	0.365	(0.009)	0.140	(0.004)	0.202
	26	167	0.383	(0.006)	0.384	(0.005)	0.383
CBDC (MAHAL.)	10	147	0.475	(0.008)	0.185	(0.003)	0.266
	26	159	0.425	(0.006)	0.428	(0.005)	0.426

evaluation substantially. Indeed, results show a significant increase in the tag coverage, combined with a slight decrease in per–song performance, when compared to the proposed class–based distance classifier (CBDC). A deeper inspection of the individual results in Figure 3.10, shows that the 18–NN (and to a lesser extent 2–NN) follows the original Ground Truth tag distribution. That is, more frequent tags are more prone to be proposed than less frequent tags. In fact, many of the less popular tags are not proposed at all, hence the lower tag coverage.

The proposed class–based distance classifier (CBDC) is able to predict all the tags in the vocabulary. CBDC can be regarded as a tag "democratizer," that is, it treats all the tags similarly, regardless of their popularity, once the centroids are computed[13]. The per–song measures for the CBDC model, specially for the one using a plain Euclidean distance, are considerably lower than the other approaches, probably due to the fact that very popular tags are not predicted as often as they were used in the original Ground Truth, a fact that can be observed in Figure 3.10.

In the following experiments, we comparatively evaluate the music annotation of our algorithms against state–of–the–art approaches. Further references to the CBDC approach will refer to the class–based distance classifier with

[13] It should be noted, however, that tag popularity does have an impact on the centroid computation.

Euclidean distance.

Comparative evaluation – Entire vocabulary. We compare the performance of our algorithm against other state of the art approaches which also used the CAL500 dataset. These approaches include the Mixture Hierarchies Estimation of Gaussian Mixture Models proposed by Turnbull et al. (2008b), the Code Bernoulli Average method presented by Hoffman et al. (2009), and the FilterBoost (an adaptation of AdaBoost) proposed by Bertin-Mahieux et al. (2008). Please refer to Section 2.5 for more details on these algorithms. An overview of the audio features used by each algorithm is presented in Table 3.19. It should be noted though, that we use the results as they were published by their corresponding authors in (Bertin-Mahieux et al., 2008; Hoffman et al., 2009; Turnbull et al., 2008b). We did not re–run nor re–implement the algorithms.

Table 3.19: Comparative description of the acoustic features used by each algorithm in the CAL500 dataset evaluation.

	Low Level					
Algorithm	Spectral	Timbre	Tonal	Rhythm	High Level	Autocorrelation
GMM		Yes				
CBA		Yes				
BOOST	Yes	Yes				Yes
DTM		Yes	Yes			
k–NN / CBDC	Yes	Yes	Yes	Yes	Yes	

For the comparison, the evaluation measures proposed in Turnbull et al. (2008b) are adopted. These measures are per–word Precision, Recall and F–measure. Per–word `Precision` and `Recall` are defined as:

$$Precision = \frac{|W_C|}{|W_A|} \qquad Recall = \frac{|W_C|}{|W_H|} \tag{3.10}$$

where $|W_H|$ is the number of tracks that are annotated with word w in the Ground Truth, $|W_A|$ is the number of tracks which the system automatically annotates with word w, and $|W_C|$ is the length of the intersection between $|W_A|$ and $|W_H|$, that is, the number of tracks for which the system correctly assigns word w.

Following the evaluation of Turnbull et al. (2008b), we annotate each track with ten words. Since the average number of tags per track is 26, the Upperbound for per–word Precision and Recall is less than 1. Table 3.20 presents the overall per–word results using all words in the vocabulary, and Table 3.21 depicts the results for each tag category. The "Random" and "Upperbound" models' results are taken from Turnbull et al. (2008b). The "Random" model samples tags, without replacement, from a multinomial distribution of the prior probability of tags, which is estimated using the tag's frequency The Upperbound model, on the other hand, uses the Ground Truth dataset for annotation. If the annotation length (number of tags per song) is fixed to some value N, we select random N tags from the Ground Truth. If the song has less than N tags, we add random tags. Since the purpose of this evaluation, as stated by Turnbull, is to verify if the systems are able to predict as many words from the vocabulary, and given the limitation of automatically annotating songs with only 10 tags, we decided to use the 2–NN and CBDC-EUC models for comparison. Additionally, we report results for the Kullback-Leibler distance based on a single Gaussian MFCC modeling. The overall results in Table 3.20 show that the CBDC algorithm outperforms the other models in recall, supporting our claim in Section 3.3, where we discussed the relevance of recall for an autotagger. The consequence of such relative good recall is also a higher F–measure for our proposed approach. The results also show that when k–NN uses a Euclidean distance over PCA components, it improves the results obtained by a Kullback-Leibler distance, but just slightly, as we could also observe in the parameter estimation step in Section 3.5.2.

Table 3.20: Overall results of the music annotation task using the CAL500 dataset. A = Annotation length, $|\mathcal{V}|$ = Vocabulary size (Turnbull et al., 2008b). GMM–MH refers to the Mixture Hierarchy Gaussian Mixture Model algorithm proposed by Turnbull et al. (2008b). CBA is the Code Bernoulli Average method proposed by Hoffman et al. (2009). Boost is the FilterBoost algorithm of Bertin-Mahieux et al. (2008). We compare against the FilterBoost using MFCC deltas as acoustic features, and the one using more features (afeats exp). Random and UpperBound are taken from Turnbull et al. (2008b). Best results are indicated in bold.

Category	$A/\|\mathcal{V}\|$	Algorithm	Precision		Recall		F-Measure
All words	10 / 174	Random	0.144	(0.004)	0.064	(0.002)	0.089
		Upper-Bound	0.712	(0.007)	0.375	(0.006)	0.491
		GMM-MH	0.265	(0.007)	0.158	(0.006)	0.198
		Boost (MFCC)	0.281	(0.066)	0.131	(0.019)	0.179
		Boost (afeats exp.)	**0.312**	(0.060)	0.153	(0.015)	0.205
		CBA	0.286	(0.005)	0.162	(0.004)	0.207
		k-NN (MFCC)	0.205	(0.008)	0.090	(0.004)	0.125
		k-NN (pca)	0.231	(0.008)	0.101	(0.004)	0.141
		CBDC	0.277	(0.008)	**0.192**	(0.007)	**0.227**

Regarding the categories in Table 3.21, our CBDC model obtained the best per–word F–measure in all but one category, instrument solo, where the 2–NN approach performed better. From the recall values for each category, we observe that our approaches do not perform so well in the instrument solo and vocal categories. The timbre information, which has shown to be very informative for instrument recognition (Agostini et al., 2003; Herrera-Boyer et al., 2003), is reduced in our case to using the mean of the first 13 MFCC coefficients for the whole audio excerpt, thus losing information about the dynamics.

Table 3.21: Results of the music annotation task, divided by categories, using the CAL500 dataset. A = Annotation length, $|\mathcal{V}|$ = Vocabulary size (Turnbull et al., 2008b). GMM–MH refers to the Mixture Hierarchy Gaussian Mixture Model algorithm proposed by Turnbull et al. (2008b). Boost is the FilterBoost algorithm of Bertin-Mahieux et al. (2008). We compare against the FilterBoost using MFCC deltas as acoustic features, and the one using more features (afeats exp). Random and UpperBound are taken from Turnbull et al. (2008b).. The best results are indicated in bold.

Category	$A/\lvert\mathcal{V}\rvert$	Algorithm	Precision		Recall		F-Measure
Emotion	4 / 36	Random	0.055	(0.012)	0.113	(0.004)	0.160
		Upper-Bound	0.957	(0.005)	0.396	(0.010)	0.560
		GMM-MH	0.424	(0.008)	0.195	(0.004)	0.267
		Boost (MFCC)	0.444	(0.025)	0.192	(0.016)	0.268
		Boost (afeats exp.)	**0.449**	(0.026)	0.176	(0.011)	0.253
		k-NN (MFCC)	0.346	(0.017)	0.146	(0.010)	0.205
		k-NN (pca)	0.397	(0.018)	0.171	(0.010)	0.239
		CBDC	0.448	(0.016)	**0.231**	(0.011)	**0.305**
Genre	2 / 31	Random	0.055	(0.005)	0.079	(0.008)	0.065
		Upper-Bound	0.562	(0.026)	0.777	(0.018)	0.652
		GMM-MH	0.171	(0.009)	0.242	(0.019)	0.200
		Boost (MFCC)	0.154	(0.024)	0.168	(0.021)	0.161
		Boost (afeats exp.)	0.236	(0.047)	0.234	(0.016)	0.235
		k-NN (MFCC)	0.152	(0.016)	0.165	(0.016)	0.158
		k-NN (pca)	0.170	(0.016)	0.189	(0.017)	0.179
		CBDC	**0.279**	(0.020)	**0.291**	(0.022)	**0.285**
Instrumentation	4 / 24	Random	0.141	(0.009)	0.195	(0.014)	0.164
		Upper-Bound	0.601	(0.015)	0.868	(0.018)	0.710
		GMM-MH	0.259	(0.010)	0.381	(0.021)	0.308
		Boost (MFCC)	0.267	(0.047)	0.363	(0.021)	0.308
		Boost (afeats exp.)	0.276	(0.044)	0.350	(0.033)	0.309
		k-NN (MFCC)	0.235	(0.017)	0.261	(0.021)	0.247
		k-NN (pca)	0.264	(0.018)	0.289	(0.021)	0.276
		CBDC	**0.323**	(0.017)	**0.445**	(0.022)	**0.374**
Instrument Solo	1 / 9	Random	0.031	(0.007)	0.155	(0.035)	0.052
		Upper-Bound	0.197	(0.019)	0.760	(0.052)	0.313
		GMM-MH	0.060	(0.012)	0.261	(0.050)	0.098
		Boost (MFCC)	0.054	(0.002)	0.374	(0.035)	0.094
		Boost (afeats exp.)	0.056	(0.001)	**0.396**	(0.017)	0.098
		k-NN (MFCC)	0.045	(0.008)	0.141	(0.029)	0.069
		k-NN (pca)	**0.076**	(0.020)	0.217	(0.042)	**0.113**
		CBDC	0.075	(0.020)	0.169	(0.038)	0.104
Usage	2 / 15	Random	0.073	(0.008)	0.154	(0.016)	0.099
		Upper-Bound	0.363	(0.014)	0.814	(0.031)	0.502
		GMM-MH	0.122	(0.012)	0.264	(0.027)	0.167
		Boost (MFCC)	0.122	(0.011)	0.239	(0.028)	0.162
		Boost (afeats exp.)	0.118	(0.007)	0.237	(0.015)	0.157
		k-NN (MFCC)	0.107	(0.013)	0.211	(0.026)	0.142
		k-NN (pca)	0.120	(0.015)	0.231	(0.028)	0.158
		CBDC	**0.148**	(0.014)	**0.307**	(0.028)	**0.200**
Vocal	2 / 16	Random	0.062	(0.007)	0.153	(0.018)	0.088
		Upper-Bound	0.321	(0.017)	0.788	(0.019)	0.456
		GMM-MH	0.134	(0.005)	**0.335**	(0.021)	0.191
		Boost (MFCC)	0.116	(0.011)	0.252	(0.029)	0.159
		Boost (afeats exp.)	0.108	(0.009)	0.228	(0.019)	0.147
		k-NN (MFCC)	0.124	(0.017)	0.224	(0.025)	0.160
		k-NN (pca)	0.147	(0.017)	0.254	(0.026)	0.186
		CBDC	**0.155**	(0.016)	0.307	(0.025)	**0.206**

A special emphasis should be given to the genre category, where our CBDC model clearly outperforms the rest of the algorithms. The k–NN models show, on the other hand, significantly lower performance than the CBDC model or the GMM-MH model, since they are focusing on the more popular tags.

Comparative evaluation – Filtered vocabulary. A closer look at the CAL500 dataset (see also Table 3.15) shows a considerable tag imbalance. While a few tags are used more than 300 times, some others are used less than 10 times. Indeed, this is a common aspect when dealing with (social) tagging (Lamere & Pampalk, 2008). Following Miotto et al. (2010), the tag vocabulary is filtered so that only tags with at least 30 positive examples are used.

This process reduces the vocabulary to 97 tags: 11 musical genres, 14 instruments, 25 acoustic qualities, 6 vocal characteristics, 35 emotions and 6 usages. The overall results for music annotation are presented in Table 3.22. Our approach is compared to the previously mentioned Mixture Hierarchies Estimation of Gaussian Mixture Models proposed by Turnbull et al. (2008b), and the FilterBoost algorithm introduced by Bertin-Mahieux et al. (2008). Moreover, other newly proposed autotagging models are also used. Content–DTM refers to the Dynamic Texture Mixture autotagger proposed by Coviello et al. (2010), which models short audio fragments as the output of linear dynamical systems. The context models, Context–DTM and Context–SVM (Coviello et al., 2011), are two–stage algorithms that use the output of a content–based autotagger as input feature vectors to model each tag in the vocabulary. These feature vectors can be regarded as Semantic Multinomials (SMNs). The rationale behind two–stage algorithms is that they explicitly tackle the problem of tag correlation. Finally, a combination of the Mixture Hierarchy GMM and the Context–DTM is also reported. The idea behind this combination, called Decision Fusion (Coviello et al., 2011), suggests that each algorithm focuses on different aspects of music. Combining such different algorithms would lead to improved results. The evaluation, in this case, is performed with 5–fold cross validation. As in the previous experiment, each track is annotated with ten words, which is still less than the average Ground Truth annotation. This issue imposes again the definition of an upper bound for per–word Precision and Recall.

Results in Table 3.22 show once more that our CBDC model has relatively the best recall, clearly outperforming the content autotaggers (except the DTM model), though only slightly better than the context models. In order to have a better understanding of the algorithms' behavior, we compare the performance of the models for the individual CAL500 tags[14]. Figures 3.7 and 3.8 present the F–measure of each tag, organized in each corresponding category. The categories are `emotion`, `musical genre`, `instrumentation`, `usage` and

[14] The Semantic Multinomials of the algorithms compared in this section were kindly provided by Emanuele Coviello.

Table 3.22: Results of the music annotation task, using a 5–fold cross validation evaluation of the CAL500 dataset. A = Annotation length, $|\mathcal{V}|$ = Vocabulary size (Turnbull et al., 2008b). The dataset is filtered by tag popularity (we consider tags that are used at least 30 times). GMM–MH refers to the Mixture Hierarchy Gaussian Mixture Model algorithm proposed by Turnbull et al. (2008b). Boost is the FilterBoost algorithm of Bertin-Mahieux et al. (2008). Context DTM (Coviello et al., 2011) is a contextual modeling of the Dynamic Texture Mixture autotagger proposed by Coviello et al. (2010) Context Fusion is a combination of different autotaggers. In this case we compare against the combination of GMM and DTM. Results are averaged from a 5-fold cross validation. Random and UpperBound are taken from Miotto et al. (2010).

$A/\|\mathcal{V}\|$	Algorithm	Precision	Recall	F-Measure
10 / 97	Random	0.231	0.101	0.140
	Upper-Bound	0.716	0.471	0.568
	GMM-MH	0.374	0.205	0.264
	Boost	0.334	0.144	0.201
	Content DTM	0.446	0.217	0.292
	Context SVM	0.343	0.223	0.270
	Context DTM	0.461	0.236	**0.312**
	Context Fusion (GMM, DTM)	**0.484**	0.230	0.311
	CBDC	0.426	**0.244**	0.310

`vocal characteristics`. Moreover, tags within each category are sorted in descending order, using our CBDC approach as a reference key to sort.

For the evaluation of the emotion category, we use the concepts of arousal and valence, based on Rusell's model of emotion (Russell, 1980). In psychology studies (Csikszentmihalyi, 1997; Frijda, 1986), arousal refers to the state of reacting to a certain stimuli. Valence, on the other hand, describes the attractiveness or aversiveness of an event, object or situation.

Our system performs comparatively well for emotions that lay on the arousal dimension —emotions such as angry, boring, calm, relax, exciting, etc.— whilst not so well for tags in the valence dimension —specially for happy, sad and their negative examples. These results confirm the findings of Laurier (2011); Yang & Chen (2011) for the specific task of mood classification from audio, where happy songs —which lay on the valence dimension— were the most difficult to classify. Furthermore, Laurier (2011) shows that spectral (complexity, kurtosis, flatness, skewness), tonal (mode) and temporal (onset, zero crossing rate) audio features play an important role on capturing the essence of emotion in music.

As for the musical genre category, no significant difference was found among the compared models. The tag filtering process has unfortunately removed 20 genres — out of 31 — from the evaluation. Empirical analysis of the Euclidean distances between the genre tags' centroids shows that tags *alternative*, *classic rock* and *rock* are very close in the 29-dimensional reduced feature representation. A plot of the first 2 PCA components of the genre tag centroids used by

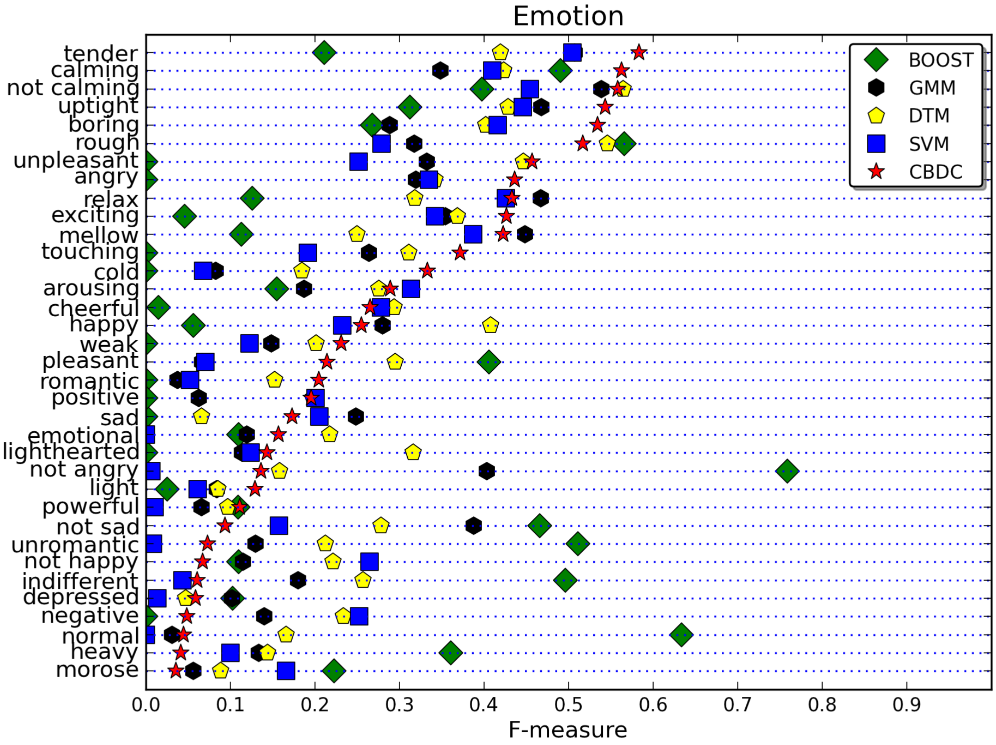

Figure 3.7: Comparative evaluation of the emotion category, using per–tag F–measure.

the CBDC model is presented in Figure 3.9. Although not synonyms, these terms share a similar semantic concept, so it makes sense that they are close together. Confusing a *rock* song with an *alternative* rock one is less harmful than confusing a *rock* song with an *r&b* one. Strict measures such as Precision and Recall, unfortunately, cannot capture these subtle semantic correlations between tags. Further reading of Figure 3.9 suggests the existence of 3 clusters of genres: {(alternative, rock, classic rock), (r&b, pop), (folk, country, jazz)}. A closer look at the contribution of all the features to the PCA components (see Appendix B.1) suggests that high level features such as party, aggressive, relaxed, fast/slow rhythm, or low level features such as Zero Crossing Rate are most prominent in the first component, which covers 22% of all the variance in the original feature representation. The latter feature, Zero Crossing Rate, can be specially relevant for distinguishing distorted music such as *metal*. The second PCA component, which covers 9% of variance, has strong loads of moods (happy), cultural (western vs. non western), rhythm (fast/slow), timbre and spectral information (energy). The X axis can probably be interpreted as timbre information, from more acoustic to more electric music. Regarding the contribution coefficients of these features in Appendix B.1, it is worth men-

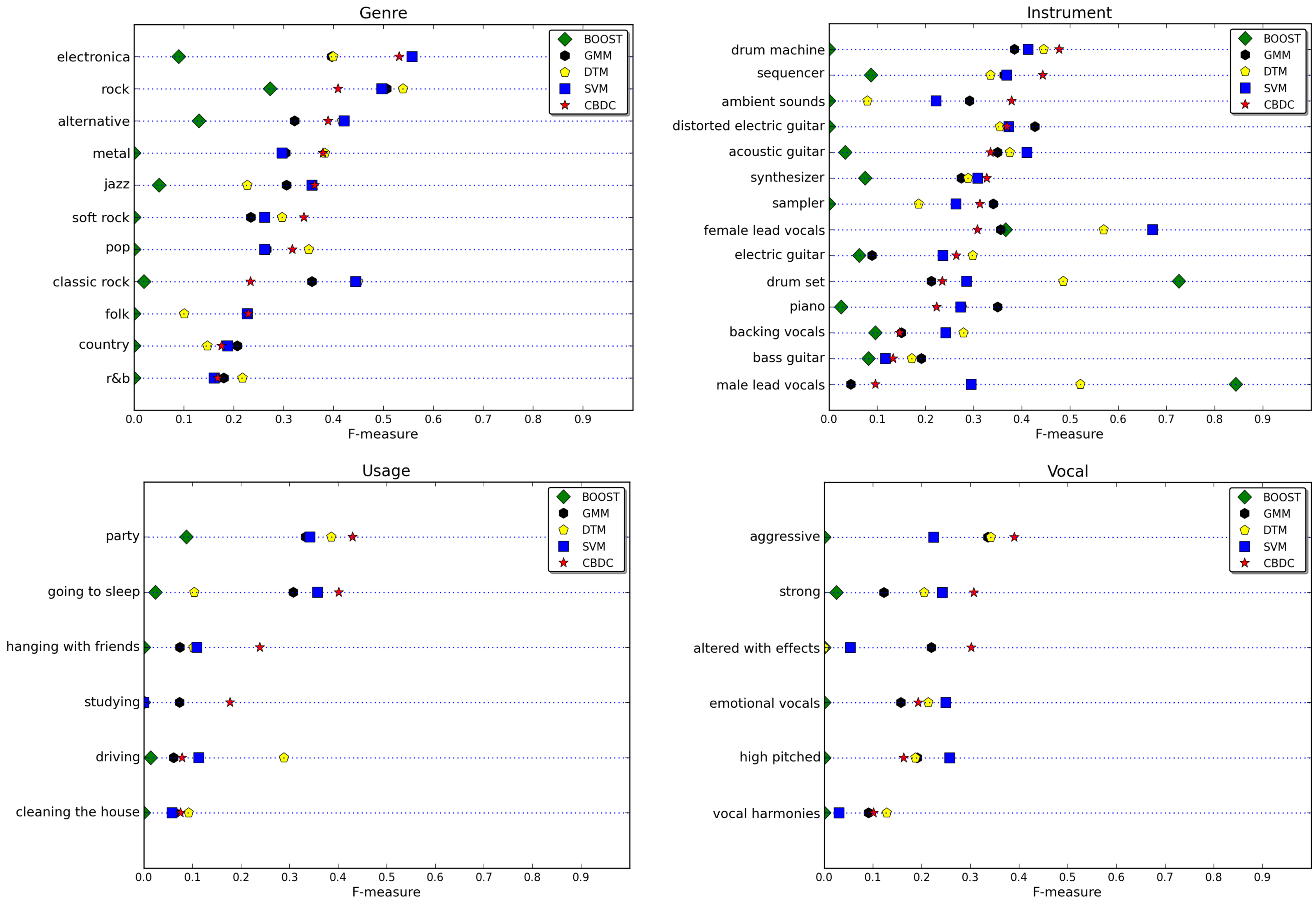

Figure 3.8: Music annotation. Comparative evaluation of the genre, instrument, usage and vocal categories, using per–tag F–measure. Our approach, CBDC, is depicted using a star symbol.

tioning that the negative coefficients can be easily interpreted for higher level features, such as moods (party Vs. not party, not relaxed and aggressive Vs. relaxed) or rhythm (fast Vs. slow). It is also interesting to see the positive contribution of western music. Indeed, The CAL500 dataset consists of 500 songs from 500 western artists (Turnbull et al., 2008b).

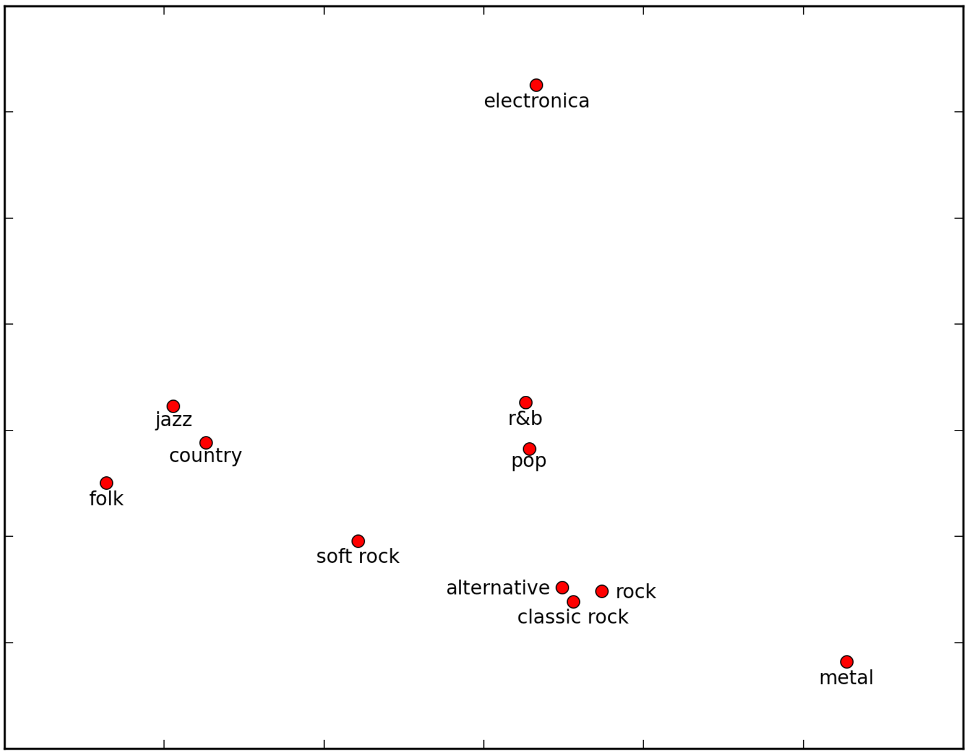

Figure 3.9: Plot of the first 2 PCA components of the CBDC model's genre tag centroids.

As for the usage category, our model outperforms the other algorithms in 4 out of 6 tags. The use of rhythm/temporal features and mood features such as aggressive or party, probably have a positive effect on this category. Some of these usage tags, such as *going to sleep*, *party* or *studying*, can be correlated to the emotion category, specially in the arousal dimension of Rusell's "circumflex model of affect" (Russell, 1980). Music to use in a *party* generally suggest more festive, exciting emotions, while music for *going to sleep* or *studying* tend to be more relaxing. The tags *cleaning the house*, and *driving* did not obtain good results. We believe that these tags are more subjective than the other usage tags. People may have a diverse range of music preferences for playing music while performing tasks such as driving or cleaning the house.

Finally, as regards to the vocal characteristics category, our model achieves

comparatively good results for tags such as *aggressive*, *strong* and *altered with effects*. The first two tags can be indirectly related to the arousal dimension of emotion, but also to perceptual features, like loudness.
It is worth mentioning the surprisingly high F–measure obtained by the FilterBoost algorithm (Bertin-Mahieux et al., 2008) for a set of tags where the other models, including ours, failed completely. However, the FilterBoost fails to predict many other tags that the rest of the models could. This may be due to the nature of boosting algorithm, where the focus goes to the weak learners with more misclassified instances.

On the analysis of per–song and per–tag evaluation measures – Filtered vocabulary. An important aspect to analyze in this section is on the use of different evaluation measures. Turnbull et al. (2008b) state that using per–song Precision and Recall can lead to artificially good results if the algorithm is only able to predict few popular tags to many songs, and ignoring the rest. While this is a key aspect for annotation —it is more relevant for a system to have as many detailed description of the music as possible— it is not answering the problem formulation correctly. In other words, music annotation refers to predicting the best classes for a given song. The element to be evaluated, thus, is a song, not (only) a tag.

Table 3.23: Overall per–song and per–word results for music annotation, using the filtered CAL500 dataset.

		Per–song measures			Per–word measures		
Algorithm	Tag coverage	Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Random	82	0.486	0.210	0.293	0.231	0.101	0.140
Upper-Bound	97	0.990	0.443	0.612	0.716	0.471	0.568
GMM-MH	97	0.377	0.158	0.223	0.374	0.205	0.264
Boost (MFCC)	76	0.677	0.292	0.408	0.334	0.144	0.201
Content–DTM	97	0.506	0.214	0.300	**0.446**	0.217	0.292
Context–SVM	93	0.404	0.170	0.239	0.343	0.223	0.270
2-NN (pca)	97	0.551	0.236	0.331	0.376	0.158	0.222
18-NN (pca)	61	**0.684**	**0.296**	**0.413**	0.299	0.147	0.197
CBDC	97	0.432	0.186	0.260	0.426	**0.244**	**0.310**

For this purpose, Table 3.23 presents per–song and per–word overall results for the different algorithms[15], using the filtered vocabulary (i.e., 97 tags) of the CAL500 dataset and a 5–fold cross validation. It is interesting to note how many systems that perform well for per–song measures do not do so well on per–word measures, and, more interestingly, vice versa. The former case seems reasonable, since autotaggers with good per–song performance (Boost, 18–NN) tend to focus on the few popular tags and thus have low tag coverage. The rest of the models (GMM-MH, DTM, SVM, CBDC) are able to predict most

[15] the semantic multinomials of the Context–DTM were not available at the time of writing this chapter.

of the tags, yet not performing so well on per–song evaluation. Content–DTM model seems to perform relatively well on both evaluations. In our case, we specially highlight the results of the 2–NN, which has a complete tag coverage, with a good per–song performance, but not very good per–tag results.

Figure 3.10 gives an insight about how the evaluated autotaggers behave when annotating songs with 10 tags. CAL500 represents the original Ground Truth tag distribution, from more popular to less popular tags. The "Upperbound" model uses the Ground truth for autotagging. Both "Random" and "Upperbound" were reproduced from the definition in Turnbull et al. (2008b).

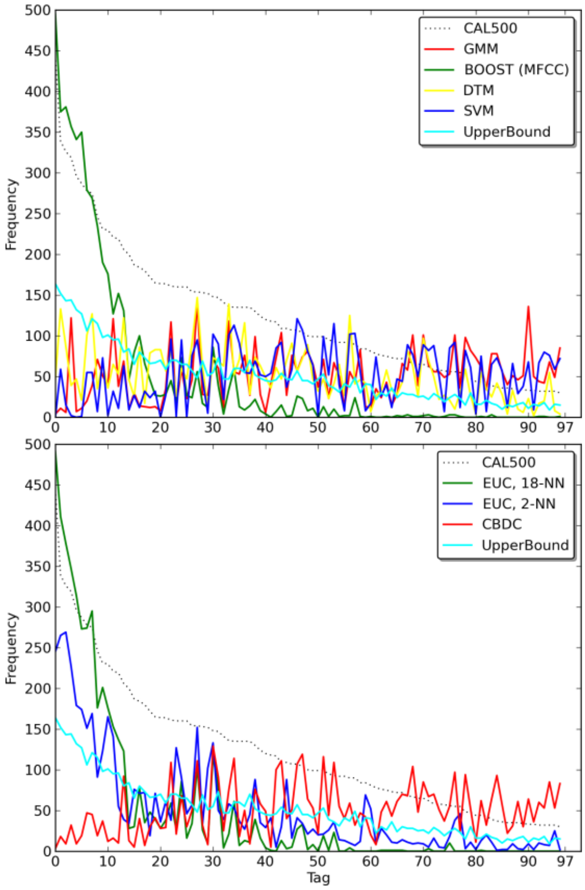

Figure 3.10: Tag distribution of the different autotaggers using the filtered CAL500 dataset. The lower figure shows the distribution for our algorithms, while the upper figure displays the distribution for the rest.

If we assume that the GT dataset is complete and consistent, then the perfect model should perform well in both per–song and per–word evaluations. As we can observe, our 2–NN approach follows a similar curve as the original dataset, yet the per–word evaluation gets worse with the less popular tags. Nevertheless, as Turnbull et al. (2008b) state, the CAL500 dataset suffers from vocabulary selection, given the fact that is done by students[16]. Indeed, the authors claim that annotations might not be of the same quality as an annotation made by musical experts, who will focus into more psychoacoustic description of music (Tingle et al., 2010). If, on the other hand, we assume that the annotation is not exhaustive and consistent, then an objective evaluation is not enough to assess the quality of our models, and therefore an alternative, subjective evaluation might be required.

Retrieval

For each query tag, we rank order its affinity to the dataset songs. In fact, this ranking is taken from the affinity of all the tags to each test song. `Mean Average Precision` (MeanAP) and `Mean Area under the ROC curve` (MeanAROC) are used as evaluation measures. Please refer to Section 2.4.1 for more details about these measures.

Table 3.24 presents experimental results for music retrieval using 10–fold cross validation and all the words in the dataset. The "Random" model is again taken from Turnbull et al. (2008b).

For the k–NN approach, the parameter configuration tested in Section 3.5.2 is chosen for music retrieval (see Table 3.17 for more information). That is, with $k = 18$, built on top of a PCA reduced data representation that keeps 75% of the original data variance (i.e. each song is represented by a vector of $\sim 29-30$ dimensions), using the highlevel descriptors mentioned in Section 3.2.1 and a Euclidean distance measure. The results show that the Mixture Hierarchy GMM is the best performing algorithm at modeling the affinity of tags, although there is no significant difference with our k–NN ($k = 18$) approach. It is interesting to note that k–NN (pca) outperforms $CBDC$ for retrieval task. This demonstrates that even though k–NN is not good enough at covering the whole tag vocabulary[17], due to popularity bias, it still shows an improvement in the affinity estimation of tags to songs.

Table 3.25 presents results for music retrieval, organized by the same categories as in the music annotation task. Our approaches still show better results in the first two categories: emotion and genre. The performance of the instrumentation category is only slightly better in MeanAROC for the CBDC and k–NN models, and in MeanAP for the Mixture Hierarchies GMM model.

3.5.4. Discussion

In this section, we have presented a thorough evaluation of our k–NN autotagging algorithm for music annotation and retrieval, using the CAL500 dataset. First, a statistical test is performed for a wide range of parameter values, in order to tune our algorithm. Table 3.17 illustrates the best configuration we

[16] who would sometimes not pay attention to the annotation.

[17] taking into account the limitation of annotating songs with only 10 tags.

Table 3.24: Overall results of the music retrieval task using the CAL500 dataset. $|\mathcal{V}|$ = Vocabulary size. GMM–MH refers to the Mixture Hierarchy Gaussian Mixture Model algorithm proposed by Turnbull et al. (2008b). Boost is the FilterBoost algorithm of Bertin-Mahieux et al. (2008). We compare against the FilterBoost using MFCC deltas as acoustic features, and the one using more features (afeats exp). Random and UpperBound are taken from Turnbull et al. (2008b). The best results are indicated in bold.

Category	$\|\mathcal{V}\|$	Algorithm	MeanAP		MeanAROC	
All words	174	Random	0.231	(0.004)	0.503	(0.004)
		GMM-MH	**0.390**	(0.004)	**0.710**	(0.004)
		Boost (MFCC)	0.305	(0.057)	0.678	(0.015)
		Boost (afeats exp.)	0.385	(0.060)	0.674	(0.010)
		k-NN (MFCC)	0.341	(0.006)	0.643	(0.005)
		k-NN (pca)	0.388	(0.007)	0.702	(0.005)
		CBDC	0.374	(0.006)	0.694	(0.004)

found for both music annotation and retrieval. That is, with $k = 18$, built on top of a PCA reduced feature representation that keeps 75% of the original data variance, using the highlevel descriptors mentioned in Section 3.2.1 and a Euclidean distance measure.

Quantitative results, however, have shown that using 18–NN for music annotation, although improving per–song evaluation, results in a lower tag coverage. In order to tackle this problem, we presented a modification of the algorithm that takes into account the tag modelization from the songs. We call this approach class–based distance classifier (CBDC). Based on centroid–based classifiers from (Han & Karypis, 2000; Kim et al., 2006; Park et al., 2003), this approach computes a centroid (cluster) for each class in the Ground Truth. A song is annotated based on a distance between the song d–dimeGaussianfeature vector (where $d = 29$ in this particular dataset) and the tag clusters. Four different distance measures were tested for the CBDC model, namely Euclidean, cosine, weighted Euclidean and Mahalanobis distance. The latter two distances can be regarded as a single Gaussian modelization of each tag. Results in Table 3.18 show that Euclidean distance has a larger tag coverage, and consequently in Table 3.20 it achieves better per–word precision and recall. This issue can be explained in two ways. On one hand, Principal Component Analysis (Jolliffe, 2002) is only applicable when model parameters are elements of a Euclidean space. If we assume that this holds for the acoustic features we use in our algorithm, then it makes sense that a Euclidean distance can achieve good results. On the other hand, a covariance matrix is built assuming that the data follows a normal (Gaussian) distribution. Nevertheless, empirical results have shown that not all the acoustic features that are used in this thesis follow a Gaussian distribution. Many other audio features ensue, for example, an exponential distribution. Examples of features with a Gaussian distribution include low level features such as *bark bands*, *skewness*, *pitch salience*, *spectral*

Table 3.25: Results of the music retrieval task, divided by categories, using the CAL500 dataset. A = Annotation length, $|\mathcal{V}|$ = Vocabulary size (Turnbull et al., 2008b). GMM–MH refers to the Mixture Hierarchy Gaussian Mixture Model algorithm proposed by Turnbull et al. (2008b). Boost is the FilterBoost algorithm of Bertin-Mahieux et al. (2008). We compare against the FilterBoost using MFCC deltas as acoustic features, and the one using more features (afeats exp). Random and UpperBound are taken from Turnbull et al. (2008b). The best results are indicated in bold.

Category	$\lvert\mathcal{V}\rvert$	Algorithm	MeanAP		MeanAROC	
		Random	0.327	(0.006)	0.504	(0.003)
		GMM-MH	0.506	(0.008)	0.710	(0.004)
Emotion	4 / 36	Boost (MFCC)	0.503	(0.031)	0.702	(0.005)
		Boost (afeats exp.)	0.478	(0.023)	0.655	(0.006)
		k-NN (MFCC)	0.460	(0.010)	0.653	(0.006)
		k-NN (pca)	**0.515**	(0.009)	**0.717**	(0.005)
		CBDC	0.452	(0.008)	0.663	(0.007)
		Random	0.132	(0.005)	0.500	(0.005)
		GMM-MH	0.329	(0.009)	0.719	(0.005)
Genre	2 / 31	Boost (MFCC)	0.094	(0.013)	0.705	(0.013)
		Boost (afeats exp.)	0.117	(0.036)	0.720	(0.011)
		k-NN (MFCC)	0.259	(0.018)	0.667	(0.013)
		k-NN (pca)	0.334	(0.017)	0.742	(0.012)
		CBDC	**0.342**	(0.016)	**0.764**	(0.011)
		Random	0.221	(0.007)	0.502	(0.004)
Instrumentation		GMM-MH	**0.399**	(0.018)	0.719	(0.006)
	4 / 24	Boost (MFCC)	0.137	(0.022)	0.707	(0.005)
		Boost (afeats exp.)	0.173	(0.030)	0.705	(0.006)
		k-NN (MFCC)	0.371	(0.018)	0.681	(0.012)
		k-NN (pca)	0.371	(0.017)	0.716	(0.012)
		CBDC	0.389	(0.016)	**0.731**	(0.011)
		Random	0.106	(0.014)	0.502	(0.004)
Instrument Solo	1 / 9	GMM-MH	**0.180**	(0.028)	**0.712**	(0.006)
		Boost (MFCC)	0.052	(0.002)	0.565	(0.025)
		Boost (afeats exp.)	0.051	(0.002)	0.650	(0.010)
		k-NN (MFCC)	0.150	(0.024)	0.577	(0.026)
		k-NN (pca)	0.180	(0.026)	0.589	(0.031)
		CBDC	0.170	(0.022)	0.635	(0.025)
		Random	0.169	(0.012)	0.501	(0.005)
		GMM-MH	**0.240**	(0.016)	**0.707**	(0.004)
Usage	2 / 15	Boost (MFCC)	0.120	(0.009)	0.621	(0.022)
		Boost (afeats exp.)	0.127	(0.007)	0.637	(0.008)
		k-NN (MFCC)	0.186	(0.013)	0.589	(0.017)
		k-NN (pca)	0.227	(0.017)	0.609	(0.019)
		CBDC	0.234	(0.016)	0.665	(0.017)
		Random	0.137	(0.006)	0.502	(0.004)
		GMM-MH	0.260	(0.018)	**0.705**	(0.005)
Vocal	2 / 16	Boost (MFCC)	0.111	(0.012)	0.652	(0.018)
		Boost (afeats exp.)	0.105	(0.012)	0.628	(0.009)
		k-NN (MFCC)	0.231	(0.020)	0.614	(0.021)
		k-NN (pca)	**0.263**	(0.021)	0.670	(0.019)
		CBDC	0.262	(0.019)	0.674	(0.018)

centroid, *spectral complexity*, *spectral decrease*, *spectral energy*, *spectral spread.* On the other hand, low level features *average loudness*, *dissonance*, *spectral flatness db spectral kurtosis*, and rhythm *first* and *second peak weights* follow

an exponential distribution. These distributions were computed using an internal test database. For more information about the features' distributions, please refer to (Freesound.org, 2011).
One way to solve this issue is by transforming the original data distribution into a normal distribution, using non linear functions such as Gaussianization (Chen et al., 2001) or box–cox (Box & Cox, 1964). These transformations, however, will change the original data representation. In our case, we decided to use PCA — a linear flattening technique — for the following reasons:

- To reduce the dimension of the data, without losing the variance of the original distribution.
- Since the PCA technique is a linear transformation of the original feature representation, adding new songs to the database or performing classification is easily achieved by simply applying the same linear transformation to the songs.
- Experimental results in this section show that this approach performs as well as or better than the state of the art algorithms.

Tables 3.20, 3.21, 3.24 and 3.25 presented comparative results of our algorithm with other state of the art approaches which used the same dataset for evaluation. Based on the obtained results, our CBDC model performs comparatively well for annotation, whereas k–NN has, on the other hand, slightly better results in music retrieval.
An additional set of experiments were carried out to analyze the effect of using per–song and per–tag evaluation measures. Results showed that many autotagging models perform well either in per–song or per–tag evaluation, not both. Further evaluation of individual tags, organized by categories, depicted the particularities of each algorithm. Our CBDC system performed comparatively well for emotions that lay on the arousal dimension —emotions such as angry, boring, calm, relax, exciting, etc.— whilst not so well for tags in the valence dimension — specially for happy, sad and their negative examples. These results confirm the findings of Laurier (2011) for the specific task of mood classification from audio, where happy songs — which lay on the valence dimension — were the most difficult to classify. As for the other categories, our CBDC system still performed as well as (genre, instrument) or better (vocal, usage) than the compared algorithms. The proposed CBDC model, altough it might seem simple and generic (a tag is represented by solely a $\sim$ 29-30 dimension vector), has revealed in experimental results that, when the audio feature representation is good enough to differentiate between classes, it can perform (in terms of per-tag evaluations) as well as or better than the state of the art approaches that use more complex, time and resource consuming algorithms. Our aim was to present additional evidence that a special care must be taken in selecting and capturing a more complete audio–related information, in order

to build successful models for automatic classification of music (Herrera-Boyer et al., 2006).
There are some cases where the models predict tags that, while not exactly the same as there were in the original annotation, are synonyms or have a semantic correlation concept. For example, given a *rock* song, if the autotagging algorithm does not predict the tag *rock*, but instead proposes *guitar*, then the objective evaluation will fail to assess the quality of this annotation. Strict measures such as Precision and Recall, unfortunately, cannot capture these subtle nuances. A human evaluation can be proposed to tackle this problem. However, it is a time and resource consuming task. Torres et al. (2007) suggest the use of vocabulary selection to limit the tags to those that are musically meaningful. Turnbull et al. (2008b) point out to using larger datasets, this time obtained through a web based game. Following a similar idea, in the next section we evaluate our models against a web–based game dataset that has been used in the MIREX evaluation fest since the beginning of the Audio Tag Classification task, the MajorMiner dataset (Mandel & Ellis, 2007).

3.6. Experiment 4: MIREX 2011

The Music Information Retrieval Evaluation eXchange (MIREX[18]) is an annual evaluation contest for Music Information Retrieval algorithms. Coupled to the International Society for Music Information Retrieval conference (ISMIR), and hosted by the IMIRSEL group[19], it aims at providing a framework with standardized datasets such that any MIR research laboratory can test their algorithms and exhaustively evaluate their results with other research teams, using community-defined evaluation metrics. The advantages of using such a contest are multiple, some of them include:

- The definition of a common Input and Output (I/O) format.
- The algorithms are compared in the same setup condition, which allows a more proper comparison.
- It favors discussion among the different research laboratories, thus reinforcing improvement of existing algorithms and the generation of new research ideas.

Several tasks have been defined within the MIREX contest, including genre or mood classification, melody extraction, tempo estimation, etc. In alignment with our research, the Audio Tag Classification task was first proposed and ran in MIREX 2008. It tests the ability of the participating algorithms to assign a variety of tags to 10–second audio clips of songs. This is "achieved" in two

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

[19] http://www.music-ir.org/

ways. On one hand, the task evaluates the binary relevance of the annotations, that is, the classification of some few relevant words to each audio clip. On the other hand, it evaluates the affinity of the audio clips to each tag in the dataset vocabulary. In order to avoid bias towards a single dataset, the task uses two different datasets for evaluation, namely the MajorMiner dataset and MIREX'09 Mood Tag dataset.

3.6.1. Dataset

MajorMiner

MajorMiner (Mandel & Ellis, 2007) is a web–based, non–paired off-line game, where a player requests a new music clip and annotates the clip with tags. The player wins points based on the originality and agreement of the tags. The MajorMiner game has collected a total of about 73000 taggings, 12000 of which have been verified by at least two users. The collection[20] used in this task consists of 2300 10-second music clips, annotated with 45 different tags, which have been verified at least 35 times, making a total of $\sim$ 9000 verified annotations. Table 3.26 gives a a description of the dataset. For further details, please refer to (IMIRSEL, 2011a; Mandel & Ellis, 2007).

Table 3.26: Additional information about the MajorMiner Dataset.

#Tracks	#Tags	Categ./Track	Tags/Track	Top Tags	Bottom Tags
2300	45	$\mu = 2.23$ $\sigma = 0.83$	$\mu = 3.82$ $\sigma = 2.07$	Drums (962) Guitar (845) Male (724) Rock (658) Synth (498)	Acoustic (40) Trumpet (39) Loud (37) Organ (35) Metal (35)

Although the tags are not organized by semantic categories, we manually assigned a facet to each tag in the vocabulary, using some thesaurus specifically built for music. The categories are gender, musical genre, instrumentation, tempo and acoustic qualities.

MIREX'09 Mood Tag dataset

Proposed by Hu et al. (2009), the MIREX'09 Mood Tag dataset is derived from mood related tags in Last.fm. The authors identified tags by using a general affect lexicon, Wordnet Affect, which is an extension of the lexical database Wordnet[21]. This extension assigns affective labels to concepts such as emotions or moods. The matched Last.fm tags in Wordnet Affect were then manually cleaned up by two human experts in Music Information Retrieval. At the end,

[20] http://www.music-ir.org/mirex/wiki/2011:Audio_Tag_Classification

[21] http://wordnet.princeton.edu/

there were 135 unique mood tags which were grouped together in 18 mood tag groups. Each audio clip can belong to multiple mood tag groups. Table 3.27 gives a brief summary of the dataset. Fore more information, please refer to (Hu et al., 2009).

3.6.2. Experimental results

In this section we present the MIREX 2011 Audio Tag Classification results. The algorithms were evaluated using a constrained (artist-filtered) 3-fold cross-validation. The constraint implies that all the songs from the same artist must appear either in the training set or the test set, not in both. This artist filtering process has shown to be of valuable importance in music similarity research (Flexer, 2007) and it is applied to avoid overfitting and bias when building the autotagging models.

Table 3.28 presents a comparative description of the acoustic features used by the different MIREX 2011 Audio Tag Classification participants. We submitted two versions of our algorithm, namely `SC1` and `SBC1`. They both share the concept of audio similarity and a final stage k–NN classifier, but differ on the way the similarity distance between songs is computed. Model `SC1` is built using the algorithm and the parameter configuration tested in Section 3.5.2. That is, the `SC1` model consists of a k–NN algorithm, with $k = 18$, built on top of a PCA reduced data representation that keeps 75% of the original data variance (i.e. each song is represented by a vector of ~ 29 dimensions), using the highlevel descriptors mentioned in Section 3.2.1 and a Euclidean distance measure. The second measure is a hybrid distance that combines the output of the first distance with a Kullback-Leibler divergence based on single Gaussian MFCC modeling, a tempo-based distance, and a semantic classifier–based distance. The latter distance component employs probability estimations of different classes of genre, mood, and instrumentation made by Support Vector Machines. For more details on the hybrid measure, please refer to Bogdanov et al. (2011).

The evaluation is divided in two different sections, corresponding to the two different aspects of audio tag classification, namely binary relevance —which measures tag classification— and affinity estimation, measuring tag ranking.

Binary relevance evaluation

In this experiment, our algorithm is evaluated on its performance at tag classification, that is, how well the model predicts a few relevant words. Tables 3.29 and 3.30 show comparative results of the MIREX 2011 Audio Tag Classification participants' performance at binary relevance evaluation, using the MajorMiner dataset and the MIREX'09 Mood Tag dataset, respectively. Results are presented as means and standard deviations computed from the 3 folds. The measures are computed on a per–song basis. Accuracy and Nega-

Table 3.27: Summary of the MIREX'09 Mood tag dataset, including the list of mood tag groups defined in (Hu et al., 2009).

Group id	Tags	#Tags	#Songs
G12	calm, comfort, quiet, serene, mellow, chill out, calm down, calming, chillout, comforting, content, cool down, mellow music, mellow rock, peace of mind, quietness, relaxation, serenity, solace, soothe, soothing, still, tranquil, tranquility, tranquility.	25	1,680
G15	sad, sadness, unhappy, melancholic, melancholy, feeling sad, mood: sad - slightly, sad song.	8	1,178
G5	happy, happiness, happy songs, happy music, glad, mood: happy.	6	749
G32	romantic, romantic music	2	619
G2	upbeat, gleeful, high spirits, zest, enthusiastic, buoyancy, elation, mood: upbeat.	8	543
G16	depressed, blue, dark, depressive, dreary, gloom, darkness, depress, depression, depressing, gloomy.	11	471
G28	anger, angry, choleric, fury, outraged, rage, angry music.	7	254
G17	grief, heartbreak, mournful, sorrow, sorry, doleful, heartache, heartbreaking, heartsick, lachrymose, mourning, plaintive, regret, sorrowful.	14	183
G14	dreamy.	1	146
G6	cheerful, cheer up, festive, jolly, jovial, merry, cheer, cheering, cheery, get happy, rejoice, songs that are cheerful, sunny.	13	142
G8	brooding, contemplative, meditative, reflective, broody, pensive, pondering, wistful.	8	116
G29	aggression, aggressive.	2	115
G25	angst, anxiety, anxious, jumpy, nervous, angsty.	6	80
G9	confident, encouraging, encouragement, optimism, optimistic.	5	61
G7	desire, hope, hopeful, mood: hopeful.	4	45
G11	earnest, heartfelt.	2	40assess
G31	pessimism, cynical, pessimistic, weltschmerz, cynical/sarcastic.	5	38
G1	excitement, exciting, exhilarating, thrill, ardor, stimulating, thrilling, titillating.	8	30
TOTAL		135	6,490

Table 3.28: Comparative description of the acoustic features used by the different participants of the MIREX 2011 Audio Tag Classification task.

Algorithm	**Spectral/ Timbre**	**Tonal Tonal**	**Rhythm/ Tempo**	**High level**	**Highlighted characteristic**
BAx	Yes				spitting method that removes irrelevant audio features
CLCB1	Yes				dynamic texture mixtures
ECL1	Yes				uses a bag of systems approach
JRx	Yes				multilabel sparse coding
TCCPx	Yes				with pretrained Universal Background Model (UBM)
PH2	Yes				PCA whitening
SSKS1	Yes		Yes		blocks of frames
SBC1	Yes	Yes	Yes	Yes	hybrid distance
SC1	Yes	Yes	Yes	Yes	

tive Example Accuracy results are also reported. Accuracy is defined as the rate of correctly predicting positive examples and ignoring negative ones. This evaluation, however, is not always reliable, specially if songs are annotated with few words, where negative examples will dominate the statistic, hence not measuring performance correctly. To overcome this issue, the MIREX evaluation members suggested the use of negative example accuracy —also known as specificity in binary classification— and positive example accuracy, which is the same as recall and it is not presented to avoid redundancy.

Our 2 participating algorithms performed third and fourth overall, out of 15 participants. Both approaches achieved practically the best recall results[22] without hampering significantly the precision. This fact is more evident in the Mirex'09 Mood Tag dataset results (Table 3.30). Although per–song evaluation measures give us a rough estimation of how the different algorithms behave, they do not allow us to compare how well the algorithms detect and classify different concepts. Thus, global measures are combined with local, per–tag evaluation measures. Figures 3.11 and 3.12 display the per–tag F–measure evaluation of the MajorMiner dataset and the Mirex'09 Mood Tag dataset, respectively.

Due to space limitations, and given that most of the research laboratories submitted two or more versions of their algorithms, we compare our approach with a subset of the best performing algorithms per each research team. In order to ease the comparison between the different participants in Figure 3.11, we group tags by similar concepts, including music genre, gender, instrumentation, tempo, perceptual characteristics and time (decades). Moreover, the tags within each group are sorted in descending order, using our approach as

[22] The recall in CLCB1 (Coviello et al., 2010) is artificial, given the low precision achieved.

Table 3.29: Comparative results (means and standard deviations) for Binary Relevance, using the MajorMiner Dataset. The best results are indicated in bold.

Algorithm	**Accuracy**	**Neg. Ex. Accuracy**	**Precision**	**Recall**	**F–measure**
BA1	0.8656 (0.0037)	0.8861 (0.0039)	0.3629 (0.0114)	0.6581 (0.0057)	0.4677 (0.0105)
BA2	0.8653 (0.0038)	0.8854 (0.0045)	0.3629 (0.0097)	0.**6621** (0.0028)	0.4687 (0.0078)
BA3	0.8640 (0.0037)	0.8841 (0.0053)	0.3598 (0.0080)	0.6604 (0.0135)	0.4656 (0.0057)
CCL1	0.7936 (0.0011)	0.8138 (0.0008)	0.2374 (0.0039)	0.5880 (0.0059)	0.3382 (0.0048)
CLCB1	0.8015 (0.0008)	0.8178 (0.0007)	0.2560 (0.0017)	0.6362 (0.0020)	0.3651 (0.0017)
ECL1	0.7942 (0.0020)	0.8142 (0.0012)	0.2388 (0.0049)	0.5915 (0.0112)	0.3403 (0.0067)
JR4	0.8565 (0.0248)	0.8969 (0.0219)	0.2995 (0.0715)	0.4427 (0.1043)	0.3558 (0.0833)
JR5	0.8500 (0.0232)	0.8896 (0.0169)	0.2872 (0.0723)	0.4465 (0.0937)	0.3493 (0.0821)
JR6	0.8800 (0.0005)	0.9101 (0.0005)	0.3863 (0.0038)	0.5742 (0.0023)	0.4619 (0.0033)
PH2	**0.9170** (0.0127)	0.9478 (0.0120)	**0.5298** (0.0670)	0.5985 (0.0711)	**0.5601** (0.0602)
SSKS1	0.9105 (0.0018)	0.**9494** (0.0013)	0.4848 (0.0109)	0.5012 (0.0083)	0.4929 (0.0095)
TCCP1	0.8608 (0.0014)	0.9230 (0.0004)	0.2064 (0.0034)	0.2100 (0.0052)	0.2082 (0.0042)
TCCP2	0.8618 (0.0021)	0.9235 (0.0006)	0.2116 (0.0054)	0.2154 (0.0085)	0.2135 (0.0068)
SBC1	0.8731 (0.0018)	0.8940 (0.0016)	0.3804 (0.0050)	**0.6613** (0.0038)	0.4830 (0.0050)
SC1	0.8712 (0.0028)	0.8925 (0.0032)	0.3749 (0.0087)	0.6545 (0.0095)	0.4767 (0.0085)

Table 3.30: Comparative results (means and standard deviations) for Binary Relevance, using the Mirex'09 Mood Tag dataset. The best results are indicated in bold. *The high recall in this case is artificial, given the low precision achieved. ** If we take into account both precision and recall, we can see that our approach achieves a good recall without hampering somehow the precision.

Algorithm	**Accuracy**	**Neg. Ex. Accuracy**	**Precision**	**Recall**	**F–measure**
BA1	0.7420 (0.0117)	0.7363 (0.0159)	0.2585 (0.0075)	0.7915 (0.0263)	0.3895 (0.0058)
BA2	0.7526 (0.0140)	0.7490 (0.0181)	0.2663 (0.0090)	0.7837 (0.0229)	0.3973 (0.0080)
BA3	0.7392 (0.0068)	0.7327 (0.0101)	0.2566 (0.0051)	0.7952 (0.0268)	0.3878 (0.0056)
CCL1	0.5090 (0.0021)	0.4741 (0.0012)	0.1517 (0.0016)	0.8109 (0.0122)	0.2555 (0.0027)
CLCB1	0.5204 (0.0021)	0.4804 (0.0012)	0.1619 (0.0025)	**0.8654*** (0.0091)	0.2727 (0.0039)
ECL1	0.5102 (0.0020)	0.4747 (0.0012)	0.1527 (0.0014)	0.8165 (0.0122)	0.2573 (0.0025)
JR4	0.7937 (0.0349)	0.8383 (0.0433)	0.2192 (0.0228)	0.3939 (0.0551)	0.2789 (0.0192)
JR5	0.8070 (0.0120)	0.8423 (0.0093)	0.2556 (0.0516)	0.4915 (0.1505)	0.3356 (0.0796)
JR6	0.7231 (0.0021)	0.7175 (0.0010)	0.2406 (0.0021)	0.7717 (0.0123)	0.3667 (0.0037)
PH2	0.8813 (0.0152)	0.9120 (0.0164)	0.4446 (0.0396)	0.6104 (0.0095)	**0.5134** (0.0275)
SSKS1	**0.9007** (0.0027)	**0.9383** (0.0009)	**0.4601** (0.0099)	0.5259 (0.0172)	0.4908 (0.0131)
TCCP1	0.8563 (0.0006)	0.9147 (0.0005)	0.2560 (0.0043)	0.2862 (0.0039)	0.2703 (0.0041)
TCCP2	0.8561 (0.0007)	0.9150 (0.0005)	0.2587 (0.0053)	0.2867 (0.0039)	0.2720 (0.0046)
SBC1	0.8483 (0.0025)	0.8708 (0.0042)	0.3703 (0.0062)	0.6550** (0.0141)	0.4730 (0.0044)
SC1	0.8501 (0.0018)	0.8737 (0.0033)	0.3723 (0.0051)	0.6460** (0.0099)	0.4723 (0.0015)

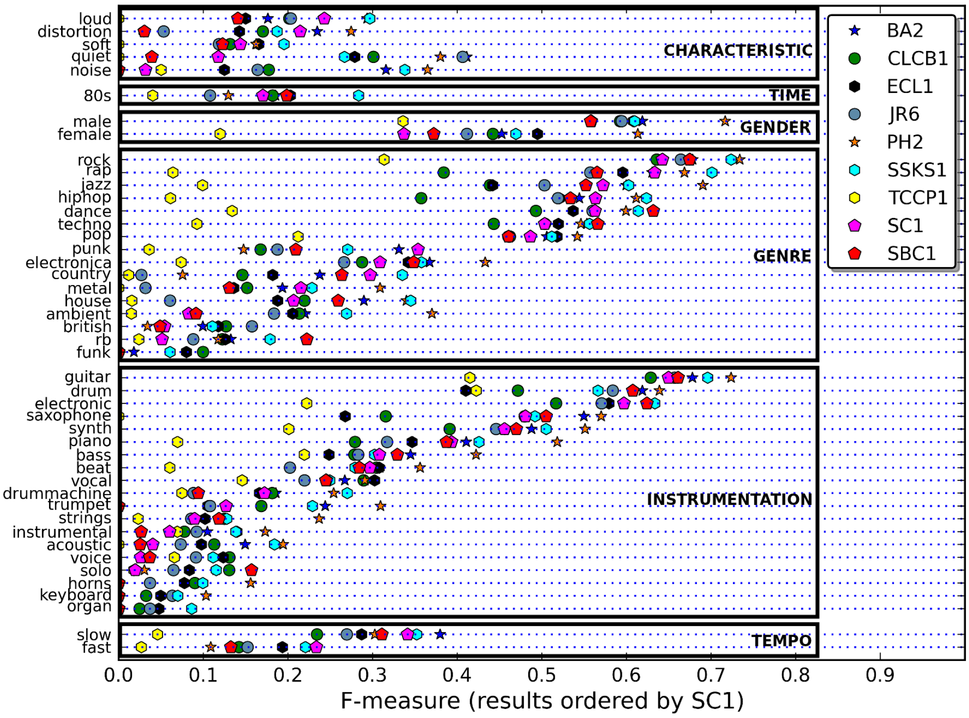

Figure 3.11: Per-tag Binary relevance evaluation of the MajorMiner MIREX 2011 Audio Tag Classification, using F–measure as a comparison metric.

a reference key to sort. A quick overview of the figure indicates that there is generally no significant difference among the participants, specially in music genre and instrumentation – which cover most of the tags in the MajorMiner dataset vocabulary. The results for the instrumentation tags seem to be proportional to the tag frequency. That is, the more positive examples of a tag in the dataset, the higher F–measure result is achieved — except for instrumental and voice tags. As regards to music genres, some results can be correlated with previous MIREX editions for the specific task of Music Genre Classification. Concretely, rap/hiphop and dance are genres that have been usually predicted accurately.

Figure 3.12 illustrates the per–tag F–measure evaluation results of the MIREX'09 Mood tag dataset, using the same subset of participants as in Figure 3.11. In this case, mood tags are merged into groups that share a semantic concept. Thus, the evaluation is not on a strictly per–tag basis, but rather on a "per–group of tags" basis. We highlight the four best performing groups using F–measure. These groups are: relax, sad, happy and angry. Interestingly, these four concepts correspond to the four core mood categories used in Laurier (2011) for the task of music mood classification. The results obtained can

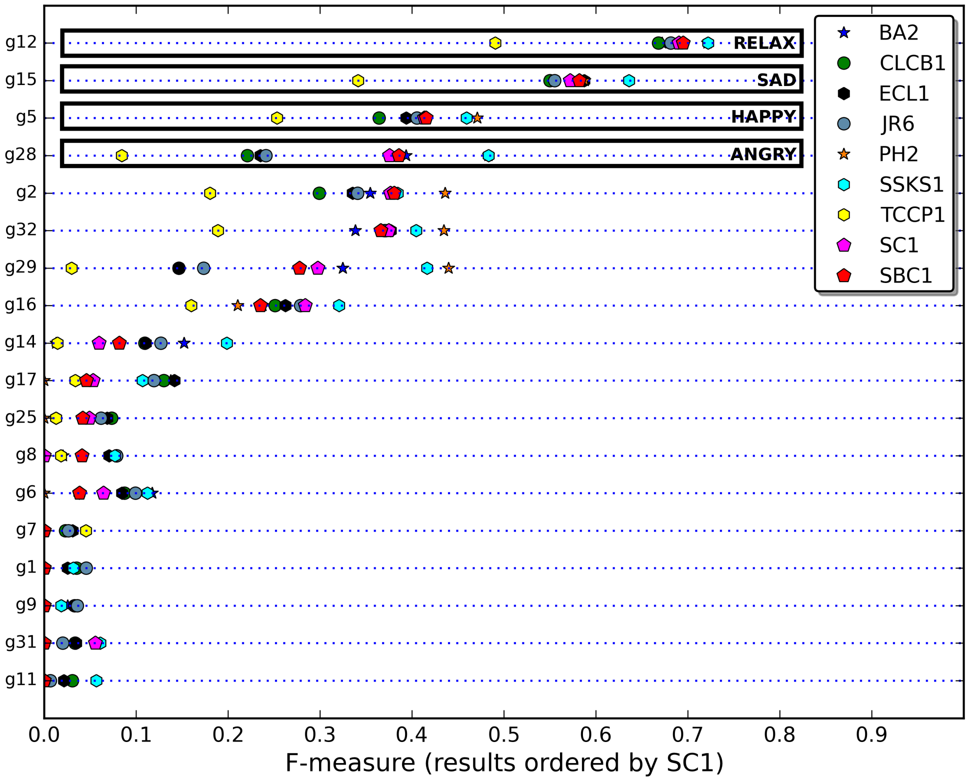

Figure 3.12: Per-tag Binary relevance evaluation of the Moods'09 MIREX 2011 Audio Tag Classification, using F–measure as a comparison metric.

reinforce the claim in Laurier (2011) that the four concepts are regarded as the basic mood categories, although the high frequency of these four tag groups (see Table 3.27) might have also had a significant impact on the final results.

Affinity estimation evaluation

The affinity estimation evaluation measures how well the models rank tags in songs, and vice versa. Tables 3.31 and 3.32 present results for the affinity estimation of tags to songs in the MajorMiner dataset and the MIREX'09 Mood Tag dataset, respectively. Results are presented as means and standard deviations computed from the 3 folds. Two different evaluation measures are used in this case: mean `Area Under the ROC` (AROC) curve and Precision–at–N (N={3,6,9,12,15}). The ROC (Receiver Operating Characteristic) curve is a plot of the True Positive Rate (sensitivity) as a function of the False Positive Rate (1-specificity) as we are moving down through the ranked list of tags. AROC is then computed by integrating the ROC curve. A random guessing would yield a mean AROC of 0.5. The Precision–at–N, on the other hand, measures whether the models rank relevant tags higher than less relevant or irrelevant ones.

The results show, again, that our two participating algorithms performed third and fourth overall. It is worth noting that our algorithm is very stable among the different folds, given the low standard deviations obtained. This is specially interesting since the first ranked algorithm, PH2, has a significantly high deviation. Indeed, looking at the original per–fold results (IMIRSEL, 2011b), it shows that the algorithm is improving systematically on average 3–4% (and 6–7% in binary relevance) in each consecutive folder.

Figures 3.13 and 3.14 report on the AROC curve results for individual MajorMiner and MIREX'09 Mood tags, respectively. For comparison purposes, we keep the same subset of MIREX participants as in the binary relevance evaluation.

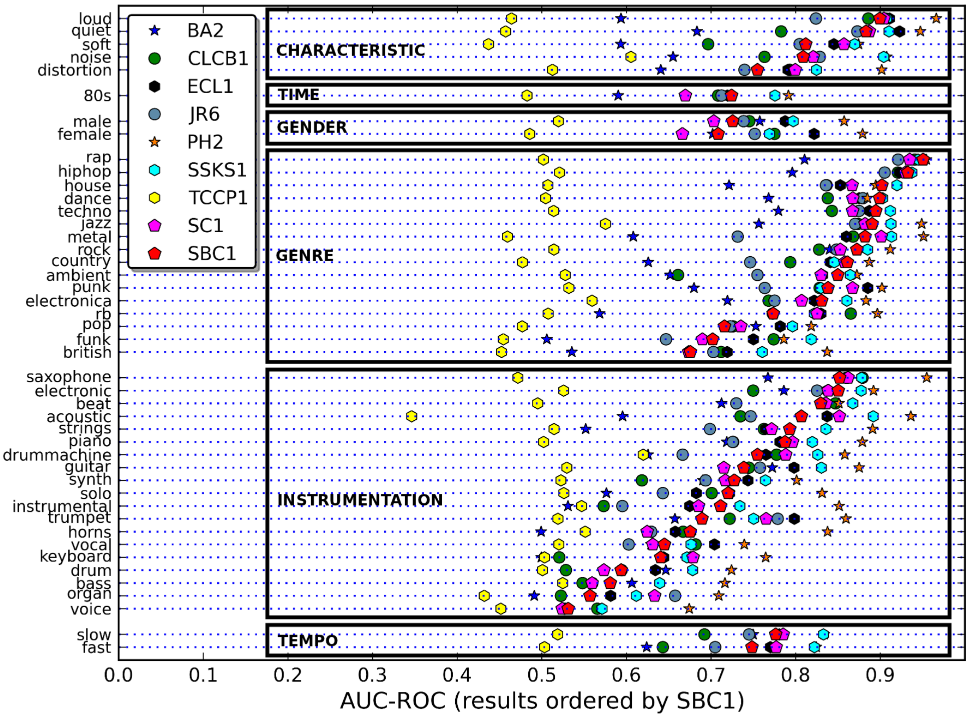

Figure 3.13: Per–tag Affinity ranking evaluation of the MajorMiner MIREX 2011 Audio Tag Classification, using Area under the ROC curve as a comparison metric.

In MajorMiner dataset, except for the TCCP1 and BA2 algorithms —which show results for most tags that are only slightly better than random guessing —, there is generally no perceived significant difference among the different models, although our approach seems to perform relatively well on genre and tempo categories. Regarding the instrumentation tags, it is worth mentioning

Table 3.31: Comparative results (means and standard deviations) for Affinity ranking, using the Major Miner Dataset. The best results are indicated in bold.

Algorithm	**AUC-ROC**	**P@3**	**P@6**	**P@9**	**P@12**	**P@15**
BA1	0.7798 (0.0039)	0.4411 (0.0061)	0.3720 (0.0107)	0.2976 (0.0041)	0.2378 (0.0043)	0.1954 (0.0033)
BA2	0.7801 (0.0017)	0.4342 (0.0123)	0.3737 (0.0070)	0.2998 (0.0035)	0.2387 (0.0032)	0.1963 (0.0033)
BA3	0.7793 (0.0058)	0.4396 (0.0086)	0.3722 (0.0092)	0.2969 (0.0053)	0.2382 (0.0044)	0.1966 (0.0040)
CCL1	0.7900 (0.0006)	0.2525 (0.0096)	0.2596 (0.0017)	0.2446 (0.0036)	0.2225 (0.0032)	0.2005 (0.0017)
CLCB1	0.8120 (0.0033)	0.3316 (0.0057)	0.2967 (0.0005)	0.2641 (0.0020)	0.2375 (0.0017)	0.2125 (0.0023)
ECL1	0.7980 (0.0019)	0.2627 (0.0080)	0.2619 (0.0003)	0.2458 (0.0052)	0.2233 (0.0039)	0.2033 (0.0027)
JR4	0.8533 (0.0007)	0.4895 (0.0063)	0.3905 (0.0046)	0.3122 (0.0043)	0.2613 (0.0028)	0.2240 (0.0018)
JR5	0.8502 (0.0018)	0.4826 (0.0082)	0.3878 (0.0045)	0.3104 (0.0035)	0.2598 (0.0047)	0.2220 (0.0028)
JR6	0.8290 (0.0016)	0.5049 (0.0005)	0.3864 (0.0039)	0.3076 (0.0024)	0.2540 (0.0021)	0.2157 (0.0017)
PH2	**0.9094** (0.0261)	**0.5902** (0.0539)	**0.4473** (0.0338)	**0.3522** (0.0202)	**0.2883** (0.0119)	**0.2439** (0.0063)
SSKS1	0.8917 (0.0026)	0.5510 (0.0104)	0.4286 (0.0054)	0.3410 (0.0062)	0.2820 (0.0037)	0.2391 (0.0023)
TCCP1	0.7942 (0.0040)	0.3783 (0.0123)	0.3100 (0.0030)	0.2688 (0.0033)	0.2336 (0.0049)	0.2013 (0.0041)
TCCP2	0.7937 (0.0042)	0.3775 (0.0114)	0.3094 (0.0026)	0.2682 (0.0036)	0.2335 (0.0050)	0.2013 (0.0042)
SBC1	0.8725 (0.0031)	0.5321 (0.0146)	0.4107 (0.0056)	0.3266 (0.0039)	0.2729 (0.0025)	0.2322 (0.0024)
SC1	0.8704 (0.0029)	0.5201 (0.0140)	0.4090 (0.0092)	0.3234 (0.0051)	0.2696 (0.0039)	0.2296 (0.0030)

Table 3.32: Comparative results (means and standard deviations) for Precision-At-N, using the Mirex'09 Mood Tag Dataset. The best results are indicated in bold.

Algorithm	**AUC-ROC**	**P@3**	**P@6**	**P@9**	**P@12**	**P@15**
BA1	0.7890 (0.0106)	0.3117 (0.0050)	0.2421 (0.0043)	0.1714 (0.0033)	0.1376 (0.0005)	0.1190 (0.0018)
BA2	0.7769 (0.0103)	0.2904 (0.0091)	0.2418 (0.0046)	0.1702 (0.0049)	0.1372 (0.0034)	0.1181 (0.0026)
BA3	0.7883 (0.0144)	0.3152 (0.0160)	0.2441 (0.0059)	0.1717 (0.0036)	0.1383 (0.0012)	0.1196 (0.0016)
CCL1	0.6945 (0.0101)	0.1896 (0.0035)	0.1879 (0.0026)	0.1607 (0.0026)	0.1370 (0.0017)	0.1189 (0.0010)
CLCB1	0.7444 (0.0096)	0.2417 (0.0136)	0.2087 (0.0062)	0.1725 (0.0035)	0.1441 (0.0019)	0.1223 (0.0013)
ECL1	0.6921 (0.0089)	0.1780 (0.0109)	0.1814 (0.0024)	0.1604 (0.0019)	0.1383 (0.0004)	0.1195 (0.0009)
JR4	0.8406 (0.0009)	0.3739 (0.0015)	0.2528 (0.0029)	0.1877 (0.0021)	0.1483 (0.0016)	0.1220 (0.0014)
JR5	0.8434 (0.0009)	0.3752 (0.0009)	0.2555 (0.0050)	0.1889 (0.0012)	0.1481 (0.0013)	0.1220 (0.0010)
JR6	0.7994 (0.0058)	0.3684 (0.0018)	0.2406 (0.0020)	0.1768 (0.0007)	0.1394 (0.0000)	0.1169 (0.0005)
PH2	**0.8739** (0.0130)	0.4033 (0.0131)	**0.2646** (0.0021)	**0.1941** (0.0005)	**0.1517** (0.0009)	**0.1236** (0.0009)
SSKS1	0.8654 (0.0018)	**0.4124** (0.0051)	0.2629 (0.0029)	0.1902 (0.0017)	0.1489 (0.0010)	0.1223 (0.0012)
TCCP1	0.8262 (0.0056)	0.3467 (0.0074)	0.2515 (0.0060)	0.1860 (0.0037)	0.1485 (0.0019)	0.1225 (0.0015)
TCCP2	0.8262 (0.0059)	0.3458 (0.0076)	0.2515 (0.0060)	0.1858 (0.0034)	0.1487 (0.0016)	0.1223 (0.0016)
SBC1	0.8448 (0.0007)	0.3876 (0.0040)	0.2573 (0.0015)	0.1895 (0.0023)	0.1498 (0.0017)	0.1231 (0.0014)
SC1	0.8455 (0.0008)	0.3887 (0.0006)	0.2582 (0.0028)	0.1901 (0.0017)	0.1499 (0.0019)	0.1231 (0.0013)

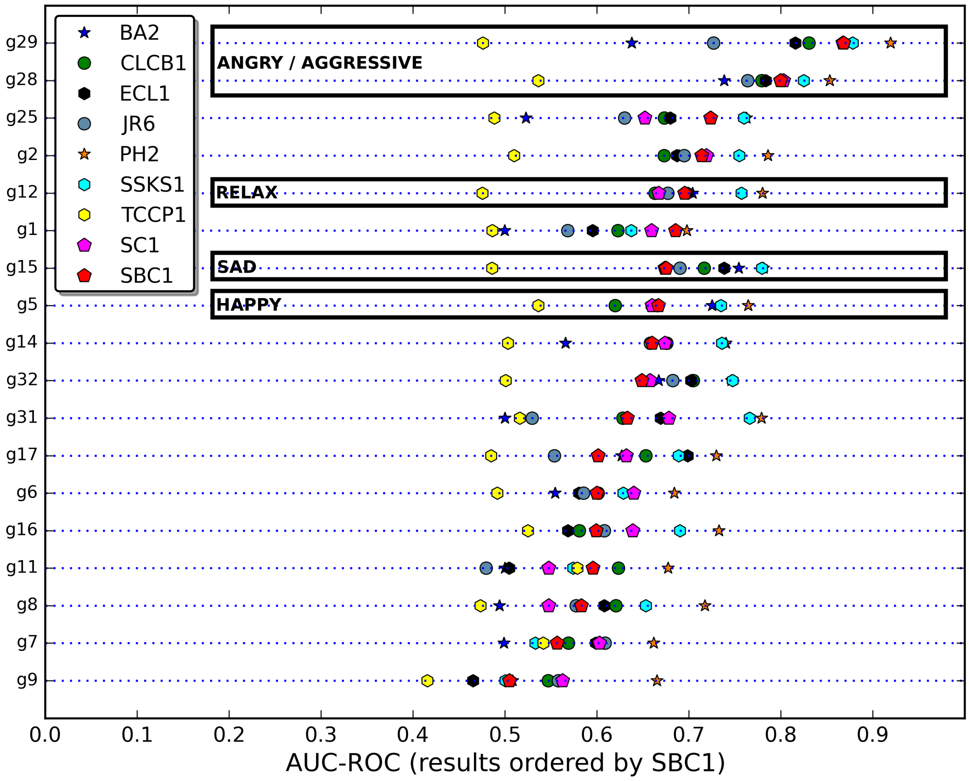

Figure 3.14: Per–tag Affinity ranking evaluation of the Moods'09 MIREX 2011 Audio Tag Classification, using Area under the ROC curve as a comparison metric.

that even if the SBC1 participation makes an explicit use of timbre–related audio features, such as MFCCs, results do not improve significantly when compared to our original SC1 approach. This again questions the usefulness of MFCCs as a (unique) feature representation of a music audio excerpt.

As for the MIREX'09 Mood tags, similar to the binary relevance evaluation, the 4 basic mood concepts rank in the top 8 mood tag groups. Our algorithms tend to perform better on the concepts lying on the arousal dimension than the valence dimension, using the paradigm introduced by Russell (1980) of classifying moods in a discrete 2–dimensional space (Csikszentmihalyi, 1997; Frijda, 1986; Juslin & Sloboda, 2001). This perception agrees with the results obtained by Laurier (2011) in the task of Music Mood tag classification, given that we are using the same library for audio feature extraction (Wack, 2011).

3.6.3. Statistical significance tests

A statistical significance test (Lehmann & Romano, 2005), also called statistical hypothesis test, is a method of making decisions using data, either from controlled or uncontrolled experiments. An experimental result is statistically significant if it is unlikely to have occurred by chance alone, according to a predetermined threshold probability, namely the significance level. The MIREX Audio Tag Classification task utilizes Friedman's ANOVA as a statistical test, along with Tukey-Kramer Honestly Significant Difference (HSD) as a post–hoc single–step comparison of significant difference among multiple evaluation means (Jones et al., 2007). The main goal is to compare the different submissions over a number of rows. The factors to be tested can be the accuracy and/or precision metrics, and a row corresponds to each tag on each fold in the 3-fold cross–validation. As stated in (IMIRSEL, 2011a), The Friedman ANOVA test should handle the variance between tags. Given the fact that the same tag can occur in different setup conditions of positive/negative examples caused by the folds, the Friedman test should replace the scores achieved by each system on each tag with their corresponding rank. Such an assumption of equal importance of rows, regardless of the unequal variance among folds, is also an often used approach at the Text REtrieval Conference (TREC) (Harman, 1993; Tague-Sutcliffe & Blustein, 1995).

Once the Friedman's ANOVA results are computed, the Tukey-Kramer Honestly Significant Difference multiple comparisons are applied. These tests are used to assess whether there is a statistically significant difference between one system and the rest.

The statistical tests' results for the MIREX 2011 Audio Tag Classification, which can be accessed in (IMIRSEL, 2011b), are summarized in Table 3.33.

3.6.4. Discussion

In this section we have presented a detailed evaluation of our submission to the MIREX 2011 Audio Tag Classification task. First, we presented an overview of the two datasets used for evaluation, namely MajorMiner and MIREX'09 Mood Tags dataset. Then, we reported the results for binary relevance (classification) and affinity estimation of tags. The evaluation was performed on a 3–fold cross–validation basis, using both per–song and per–tag measures.

Our algorithm achieved a good overall rank (between second and fourth) in almost all the experimental results. It is worth mentioning the high recall achieved in the binary relevance (classification) evaluation. This coincides with the claim we made in the first experiment (Section 3.3.2) that recall is more informative in our case, since we are interested in knowing how well the algorithm can propose all the relevant tags. Indeed, we believe that recall is slightly more important than precision in the audio tag classification task. This is explained by the fact that music collections for autotagging are usually

Table 3.33: Statistical significance tests for the MIREX 2011 Audio Tag Classification task.

Eval. measure	**Dataset**	**Statistical significance**
Binary relevance		
F-measure by Fold	MajorMiner	—
	Mood Tags	—
F-measure by Tag	MajorMiner	$(PH2, SSKS1, BAx) > (SC1, SBC1)$ $SC1, SBC1 > TCCPx$
	Mood Tags	—
Affinity estimation		
AUC-ROC by Fold	MajorMiner	—
	Mood Tags	—
AUC-ROC by Tag	MajorMiner	$PH2 > SBC1$, $PH2 > SC1$ $SBC1 > (JR4-5, BAx, TCCPx)$, $SC1 > (JR5, BAx, TCCPx)$
	Mood Tags	$PH2 > SBC1$
Precision at N	MajorMiner	—
	Mood Tags	—

weakly labeled. That is to say, the absence of a particular tag annotation for a given excerpt does not necessarily mean that the tag is not relevant to the excerpt. Moreover, proposed results with high recall can be filtered out in a post-processing step using additional information, e.g., contextual information, which can improve precision significantly.

Besides the good qualitative results obtained in this experiment, we specially highlight the following characteristics of our approach:

1. It is fast. Using a server machine with an Intel Xeon 4–core CPU and 8GB of RAM, it took, on average, 32 seconds to train and classify a fold of 1533 songs for training and 767 songs for testing.

2. It is scalable. In our original approach, `SC1`, for example, each audio excerpt is defined by a single vector of 29-30 dimensions.

3. It is easy to implement. Built on top of a k–NN classifier, our approach defines simple heuristics for classification and affinity.

4. It is consistent. The overall results, averaged per fold, show a very low standard deviation (see, for instance, how does PH2 perform in different folds (IMIRSEL, 2011b)).

The MIREX 2011 Audio Tag Classification task also includes a set of statistical significance tests, in order to check whether the practical difference of results is statistically significant or not. Statistical significance was found mainly in

MajorMiner at tag level, probably due to the diverse concepts used in the tag vocabulary. Yet, in many evaluation results there was no statistical significance. This issue is not new in MIREX Audio tag classification. Indeed, very few algorithms have shown to be statistically significant, which raises an open research question: are all the evaluation measures used in the statistical tests good and sufficient to discriminate between different algorithms? Or, on the hand, are we actually reaching a "glass ceiling" (Aucouturier & Pachet, 2004) for audio tag classification, even with novel algorithms that take into account correlations between higher level concepts?

Regarding the evolution of results in the MIREX Audio Tag Classification, if compared with those from 2010's edition [23], the results of the MIREX 2011 edition show a slight improvement. In our case, we got the same results as the best performing algorithm in 2010.

Results showed that a simple model using a variety of audio features, ranging from low level (spectral, timbre, pitch,etc), tonal, temporal and high level, can achieve as good results as or even better results than many other models that use much more complex, time and resource consuming algorithms. Interestingly, these aforementioned models rely on spectral and/or timbre information only. We believe that these results are yet another additional evidence that special care must be taken in selecting and capturing a more complete audio–related information, in order to build successful models for automatic tagging of music.

3.7. Experiment 5: iTMS-500K dataset

3.7.1. Dataset

The last dataset is a large collection consisting of almost 550,000 annotated songs. The tags were collected from the Last.fm[24] repository during March, 2009. Each audio–tag pair has an associated weight, a value in the range 1–100, which represents the relevance of the tag to the corresponding song[25]. The original (noisy) dataset contained 261,603 different tags, making a total of 3,705,566 annotations. However, many of the these tags were misspellings (for example: *hip-hop*, *hip hop* and *hiphop*), or poorly and rarely used tags. Hence, we proceeded to clean the dataset by following the next steps:

- Removing spaces and special characters. For example, *hip-hop*, *hip hop*, and *hiphop* are all converted into one tag: *hiphop.*

[23] http://www.music-ir.org/mirex/wiki/2010:MIREX2010_Results

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

[25] The exact formula is a trade secret of Last.fm, but it takes into account how many times that tag has been applied to the song.

- Merging tags and their weights. If a song has tag *hip-hop* with weight 66 and *hip hop* with weight 40, the merged tag *hiphop* will have a weight of $66 + 40 = 106$ (though we limit it to 100, that is the maximum weight.
- Filter by tag weight to remove poorly associated tags. We only consider tags in a song with a weight ≥ 20.
- Filter by tag popularity. We keep tags that were used at least 10 times.

Finally, we ended up with a dataset of 546,386 tracks and 26,324 unique tags. Table 3.34 shows some additional information about the dataset, while Figure 3.15 displays the distribution of tags in this dataset. The best fit of the tag distribution is obtained with a power law function $x^{-\alpha}$, with parameter $\alpha = 1.951939$. It is worth noting that from the total 26324 tags, 82 are used more than 5000 times, whilst 12314 tags (which represents roughly the 47% of all the tags) are used less than 20 times. Then, Figure 3.16 depicts a tag cloud of the 500 most frequent tags in the iTMS-500k dataset. As one can observe, most of these tags are related to musical genre, which coincides with the findings of Lamere (2008).

Table 3.34: Additional information about the iTMS-500k Dataset. Tags were gathered from *Last.fm* during March, 2009.

#Tracks	#Tags	Tags/Track	Top Tags	Bottom Tags
546386	26324	$\mu = 5.52$	Rock (94765)	Flower (10)
		$\sigma = 5.14$	Alternative (44796)	Bathory (10)
			Pop (43624)	Crushing riffs (10)
			Indie (43271)	Great drummer (10)
			Jazz (40246)	Bag of tricks (10)

3.7.2. Experimental results

In this section, we evaluate our system using a considerably large music collection. Our goal is to carry out the evaluation qualitatively and quantitatively. In order to have a sense of performance quality, we compare our results to random baseline and one of the representative state–of–the art autotagging algorithms, proposed by Mandel & Ellis (2008). The latter approach, as described in Section 2.5, trains tag models using a one–versus–all Support Vector Machine algorithm over a set of audio features, including Mel Frequency Cepstral Coefficients (MFFCs) and rhythm descriptors (Mandel & Ellis, 2008) [26]. An audio excerpt is finally represented by a 380–dimensional feature vector. The algorithm uses Platt et al. (1999) scaling to convert SVM decision function

[26] the code for this algorithm was kindly provided by Michael Mandel in April 2010, http://www.music-ir.org/mirex/abstracts/2010/MP1.pdf.

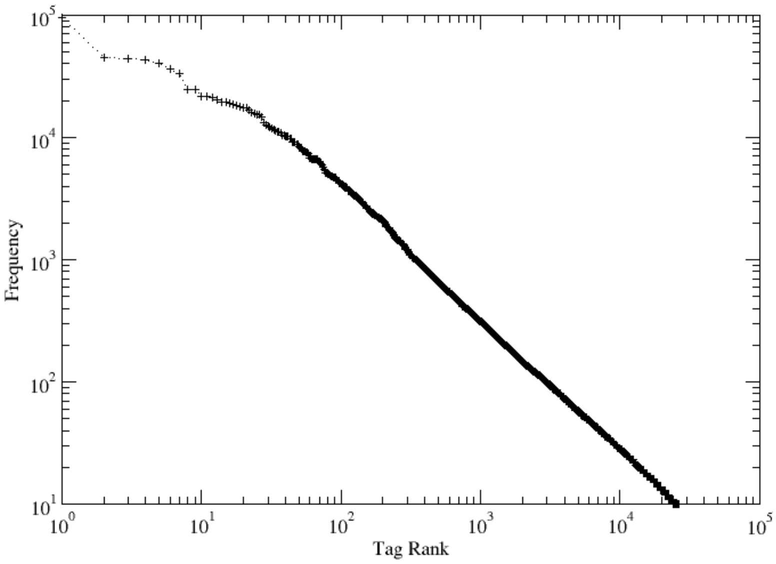

Figure 3.15: A log–log plot showing the tag distribution in the iTMS-500K dataset. The curve follows a power law distribution, with $\alpha = 1.951939$.

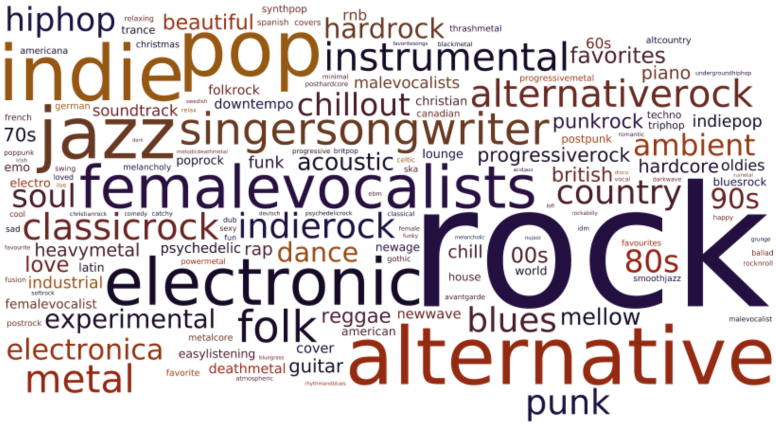

Figure 3.16: A tag cloud of the top 500 tags in the iTMS-500k dataset.

scores to probabilities so that tag relevance can be compared across multiple SVMs. Besides the standard, qualitative evaluations of classification and affinity ranking, we additionally report on quantitative results for both algorithms. We perform tests to assess the scalability of these models, in terms of CPU time.

Quantitative evaluation

Several experiments were conducted to assess the computational cost of training and classifying a large music collection. CPU time is used as a factor for comparison. Each experiment consists of a different dataset size. That is, we want to figure out whether the models scale linearly in time, while increasing the dataset size, let us call it $\mathcal{D}$. The following dataset sizes were defined: {500, 1000, 2000, 5000, 10,000, 20,000, 50,000, 100,000, 200,000, 546,386}. A 5–fold cross validation is used to test each experiment — which means 80% training and 20% testing. We compute the CPU time of each fold iteration and take the mean as the representative CPU time for each experiment. This time includes training models and ranking all the tags in the dataset.

All the experiments were conducted using a server machine with Intel Xeon 4–core CPU and 8GB of RAM, running Debian Linux 64-bit as an Operating System. Figure 3.17 depicts a lin–lin plot of the CPU time required by both models for each dataset size. Results for the SVM model (Mandel & Ellis, 2008) are reported up to a dataset size of 50,000 audio excerpts, since the machine unfortunately ran out of memory with a 100,000 excerpts dataset.

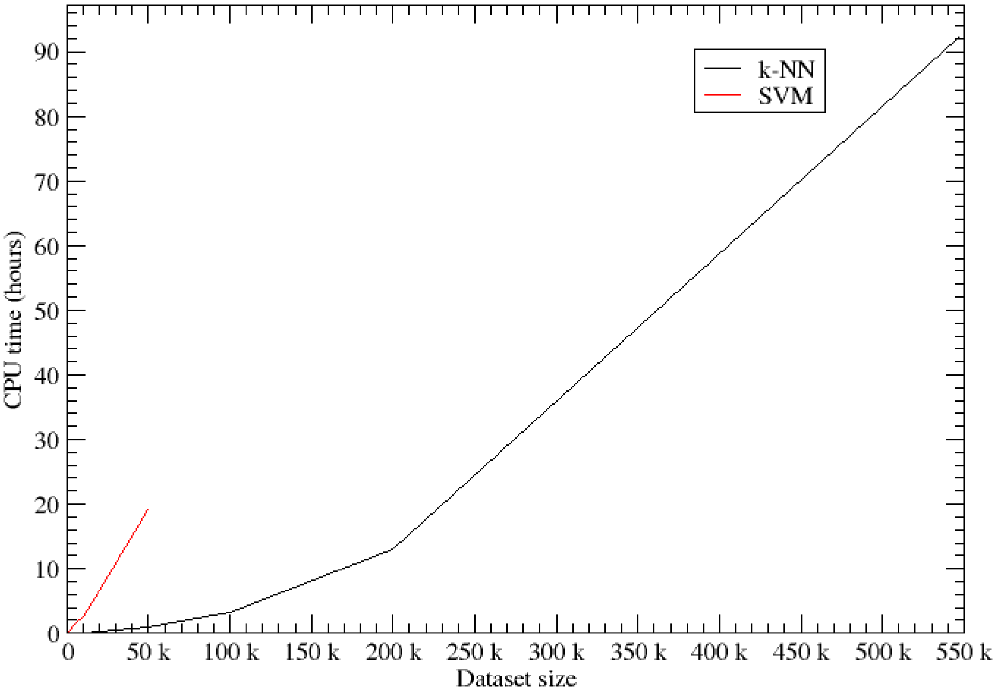

Figure 3.17: A lin–lin plot of the time consumed by the two compared algorithms for learning and ranking all the tags in the test set. The results are means taken from a 5–fold basis.

For $\mathcal{D} = 50,000$, our algorithm required, on average, 61 minutes for performing feature selection and learning from a training set[27] of 40,000 audio excerpts, and finally classifying a test set of 10,000 audio excerpts. In the case of Mandel's SVM algorithm, it took, on average, 1142 minutes for the same experiment, ~ 19 times slower than our approach. It should be considered, though, that the authors use the LibSVM library, which has shown to have a very high computational cost in large databases, especially when the training strategy for multiclasses is one–versus–all (Dong et al., 2005). Other techniques have been proposed to adapt SVMs in order to handle very large datasets (Dong et al., 2005; Tsang et al., 2006).

Our algorithm uses an efficient implementation of PCA and k–NN (Wack, 2011). In order to apply dimension reduction, the algorithm proceeds by splitting the collection in random sub-datasets of 30,000 audio excerpts. It applies PCA over one dataset, and then uses the same linear transformation for the remaining sub-datasets.

It should be noted that the classification step includes the affinity ranking of all tags in the training dataset. In order to compute this ranking using our memory–based model (k–NN), each test query needs to iterate over $\mathcal{R}$ results, where $\mathcal{R}$ is the size of the training dataset. For instance, when $\mathcal{D} = 50,000$, with $\mathcal{R} = 40,000$ audio excerpts for training and $\mathcal{T} = 10,000$ for testing, our algorithm needs $\mathcal{R} \times \mathcal{T} = 40,000 \times 10,000 = 4.0 \times 10^8$ iterations. The entire iTMS-500K dataset – which consists of 546386 audio excerpts — requires $437109 \times 109277 = 4.8 \times 10^{10}$ iterations. If our goal is to annotate a song with a few relevant tags, then the number of operations (the complexity) depends on the number of similar songs we chose, that is to say, the k in a k–NN classifier, and it would scale linearly with the number of testing instances.

Qualitative evaluation

This section presents experimental results on audio tag classification, using similar evaluation measures as in previous experiments. A constrained 5–fold cross–validation is performed for each dataset size as defined in Section 3.7.2. The constraint implies that tracks from the same artist must appear either in the training set or the testing set, not both. This artist filtering process has shown to be of valuable importance in music similarity research (Flexer, 2007) and it is applied to avoid overfitting and bias when building the autotagging models. Our system is again built using the algorithm and the parameter configuration tested in Section 3.5.2, and illustrated in Table 3.17. That is, the model consists of a k–NN classifier, with $k = 18$, built on top of a PCA reduced feature representation that keeps 75% of the original audio features variance (i.e. each song is represented by a vector of ~ 29 dimensions). The audio features include the high level semantical descriptors mentioned in Sec-

[27] A memory–based algorithm, such a k–NN, does not require a training step, but instead it postpones the training to when a test query is presented.

tion 3.2.1. Finally, similarity distance between audio excerpts is computed via a Euclidean distance measure.

Table 3.35: Comparative results for audio tag classification using the iTMS-500K dataset and per–song evaluation measures. The best results are indicated in bold.

Dataset Size	Algorithm	Vocabulary	Tag coverage	Precision	Recall	F-Measure
500	Random	159	159	0.026	0.064	0.037
	SVM		156	0.065	**0.193**	0.098
	2–NN		159	0.104	0.182	0.133
	18–NN		63	**0.216**	0.169	**0.189**
1000	Random	325	325	0.015	0.033	0.020
	SVM		275	0.061	0.154	0.087
	2–NN		324	0.088	0.154	0.113
	18–NN		101	**0.221**	**0.159**	**0.185**
2000	Random	628	628	0.010	0.017	0.013
	SVM		472	0.037	0.079	0.050
	2–NN		627	0.077	0.120	0.094
	18–NN		147	**0.229**	**0.136**	**0.171**
5000	Random	664	664	0.009	0.015	0.011
	SVM		575	0.064	0.144	0.089
	2–NN		664	0.084	0.132	0.103
	18–NN		212	**0.233**	**0.150**	**0.182**
10,000	Random	1340	1340	0.005	0.008	0.006
	SVM		1016	0.044	0.090	0.059
	2–NN		1327	0.074	0.108	0.088
	18–NN		287	**0.233**	**0.134**	**0.170**
20,000	Random	2766	2766	0.003	0.004	0.003
	SVM		2024	0.033	0.065	0.044
	2–NN		2708	0.068	0.098	0.080
	18–NN		378	**0.233**	**0.125**	**0.163**
50,000	Random	6326	6326	0.001	0.001	0.001
	SVM		3712	0.030	0.054	0.039
	2–NN		6205	0.064	0.086	0.073
	18–NN		556	**0.235**	**0.117**	**0.157**
100,000	Random	10837	10837	0.001	0.001	0.001
	SVM		N/A	—	—	—
	2–NN		10683	0.062	0.084	0.071
	18–NN		699	**0.238**	**0.114**	**0.154**
200,000	Random	17848	17848	0.000	0.001	0.000
	SVM		N/A	—	—	—
	2–NN		17605	0.063	0.088	0.073
	18–NN		874	**0.243**	**0.116**	**0.157**
546,386	Random	26324	26324	0.000	0.000	0.000
	SVM		N/A	—	—	—
	2–NN		25923	0.062	0.093	0.075
	18–NN		1026	**0.248**	**0.132**	**0.172**

Our algorithm is evaluated on its performance at tag classification, that is, how well the model predicts a few relevant words for each audio excerpt. The results are compared with the SVM model, proposed by Mandel & Ellis (2008). Table 3.35 presents the comparative results using different — and growing —

dataset sizes, taken from the original iTMS-500K dataset. The evaluation measures, which are standard IR measures, are computed on a per–song basis. Additionally we report the tag coverage of each model. Tag coverage captures information about the number of tags that were predicted by the different models [28].

Unfortunately, we found that the compared models have different tagging behavior. While the SVM model proposed by Mandel & Ellis (2008) for instance returns a considerably large list of predicted tags, the 18–NN model predicts fewer tags per test song, probably due to the restrictions imposed by the voting threshold. Based on these findings, we also report the results for a less restrictive 2–NN model. Indeed, the voting threshold does not have any effect when $k \leq 2$, since $0.2 \times 2 = 0.4 \simeq 0$[29], which means that each test track is annotated with all the tags from the 2 nearest neighbor, without any filtering, that is: $\tau(s) = \tau(n_1) \cup \tau(n_2)$. With the aim of comparing the different models, we impose a limitation of top ten tags per song. Following Turnbull et al. (2008b), the "Random" model samples tags, without replacement, from a multinomial distribution of the prior probability of tags, which is estimated using the tag's frequency.

The results show that the 18–NN model outperforms the other two algorithms in terms of per–song F–measure, in all the dataset sizes. Nevertheless, the tag coverage (number of predicted tags) is considerably low. These results agree with those obtained in Section 3.5.3 for the CAL500 dataset. The 2–NN model predicts many of the tags in the original Ground Truth dataset, consequently decreasing in per–song performance, still the results are slightly better than the SVM model of Mandel & Ellis (2008). Results for the latter method are only reported up to a dataset size $\mathcal{D} = 50,000$, since the server machine unfortunately ran out of memory with a 100,000 excerpts dataset.

Table 3.36: Summary of predicted tags by algorithm for the iTMS-500K dataset.

		Per–word F–measure > 0.1		
Dataset size	Vocabulary	SVM	2–NN	18–NN
500	159	44	51	30
1000	325	49	75	43
2000	628	51	70	42
5000	664	86	80	56
10,000	1340	93	83	61
20,000	2766	111	92	61
50,000	6326	128	102	74

As for the per–tag evaluation, Table 3.36 presents a summary of the number of predicted tags by algorithm. In this experiment, a tag is considered to be

[28] It should be noted though that the term predicted does not mean accurately predicted.

[29] The rounding has also an effect on the algorithm's voting.

predicted if the per–tag F–measure of that tag is greater than a threshold of 0.1. It is interesting to note the significantly low number of predicted tags. In this case, Mandel & Ellis (2008) is slightly better than our models. This is probably due to the nature of such an unbalanced dataset, where 82 tags are used more than 5000 times, while 12314 tags are used less than 20 times. In order to have a thorough look at the predicted tags, Figures 3.18 and 3.19 depict a comparative per–tag evaluation of the three compared models, for genre and the rest of categories, respectively. Musical genre is indeed the most prominent music concept/facet used in social tagging web sites such as Last.fm (Lamere, 2008).

In Figure 3.18, tags are sorted from the best predicted genre tags using the SVM model of Mandel & Ellis (2008) to the best performing genre tags with our 18–NN model. As one can observe, there are some genres where all the models perform well, namely *rap*, *hiphop* and ambient. These results agree with the MIREX audio tag and genre classification tasks (IMIRSEL, 2011b), showing for instance that *rap/hiphop* is the easiest model to learn. The SVM model tends to perform well on genres that might benefit from rhythm information, such as *house*, *trance*, *drum and bass* or *reggaeton*. On the other hand, the 18–NN model's predictions cover a variety of music genres, from more acoustic (*country*, *folk*), to more electric (*alternative*, *metal*, *death metal*); from relax music (*chillout*) to more party music (*dance*, *pop*).

Appendix B.1 shows a list of audio features that contribute more to the first two PCA components (which already count for the 33% of the original feature variance). This list includes high level features related to mood (party vs. not party, relaxed, aggressive, party), rhythm (fast, slow), and a variety of musical genres, which can confirm the results obtained for music genres in Figure 3.18. The list also includes information about tonal, timbre and the distinction between voice and instrumental music. The latter might have influence on the instrumentation facet in Figure 3.19.

3.7.3. Discussion

The aim in this last experiment was to evaluate our autotagging algorithm with a very large dataset, both quantitatively and qualitatively. We addressed this experiment by performing evaluations with different and growing dataset sizes. In order to assess the quality of our results, we compared them against a representative algorithm in the state–of–the–art (Mandel & Ellis, 2008). This approach uses ν–SVM (Schölkopf et al., 2000) as a Machine Learning algorithm, included in the LibSVM library (Chang et al., 2001).

The quantitative evaluation consisted in capturing the time complexity of each algoritm. We monitored the CPU time needed by each algorithm for training and ranking all the tags in the dataset. Results for SVM added more evidence that classic SVM algorithms cannot scale to large data. Our implementation, on the other hand, scaled reasonably well for a half a million song dataset.

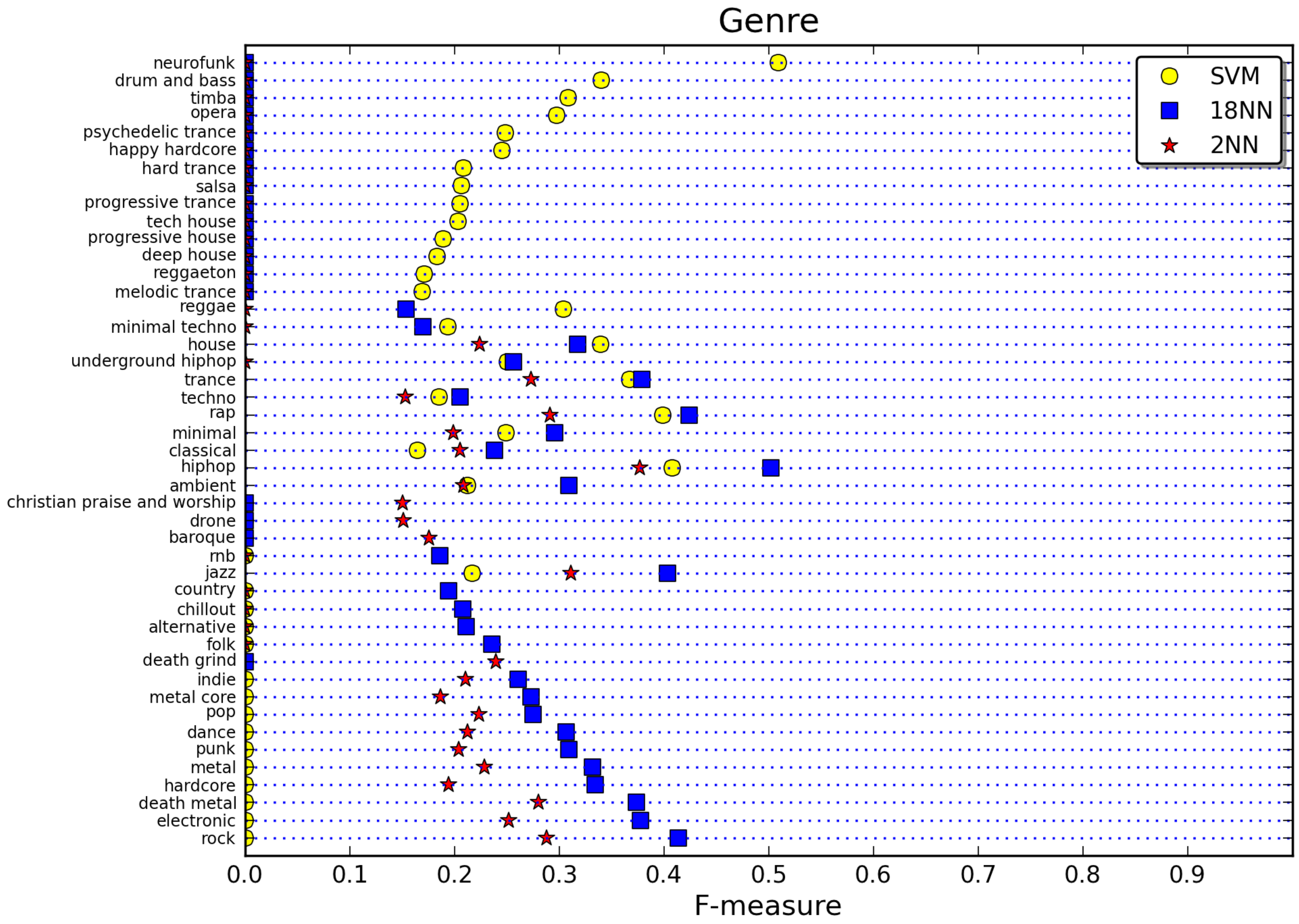

Figure 3.18: Comparative evaluation of the iTMS-500K dataset, using per–tag F–measure. Results are reported for a subset of the genre category.

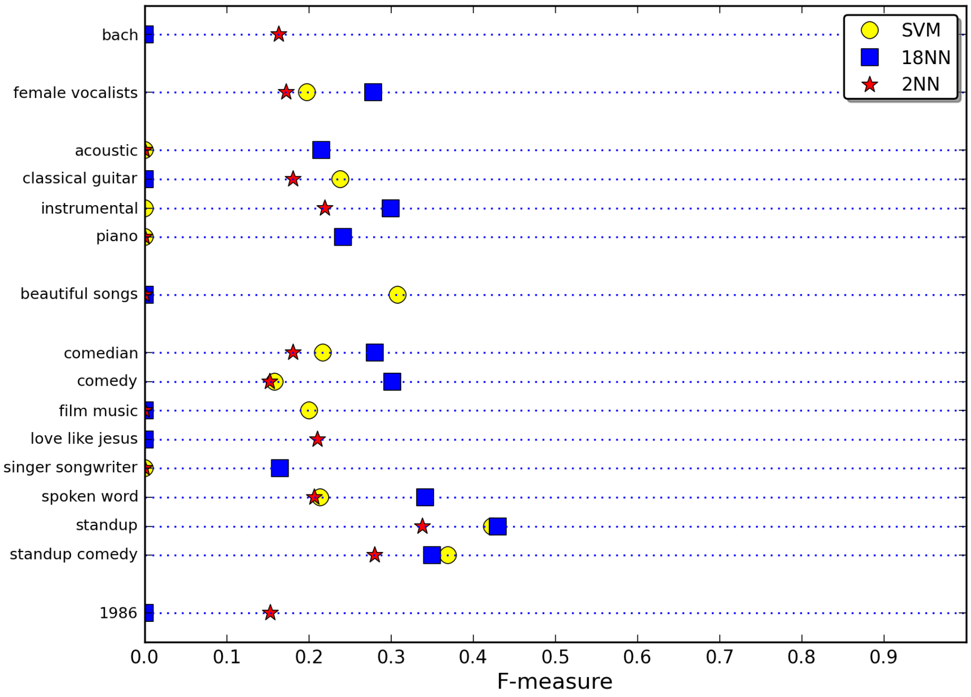

Figure 3.19: Comparative evaluation of the iTMS-500K dataset, using per–tag F–measure. Results are reported for the rest of categories.

Several experiments were performed for qualitatively evaluating our approach, using both local (per–tag) and global (per–song) evaluation measures. The results showed that our models outperform Mandel & Ellis (2008) in per–song evalations, while the latter performs slightly better in per–tag evaluation. A clooser look at the predicted tags indicates that each model is good at predicting different tags, belonging to the same or to different music facets. In musical genre, Mandel & Ellis (2008) performed releatively well for *house*, *trance*, *drum and bass* or *reggaeton*, where rhythm is more prominent. Our k–NN model, on the other hand, was able to predict a variety of other music genres, ranging from acoustic to electric, from relax to party or aggressive music.

Yet, the generally low results achieved in this experiment might suggest that the task at hand is very difficult to solve. Even though the original Last.fm tags were cleaned and filtered, the tag frequency follows a power law distribution (see Figure 3.15). While 82 tags are used more than 5000 times, 12314 tags (which represents roughly the 47% of all the tags) are used less than 20 times. The dataset also suffers from weak labeling. If a tag is not used in a song it does not necessarily mean that this song cannot be associated with that tag. Moreover, some tags are synonyms (e.g., *vigorous* and *energetic*) or share concepts (e.g., *rock* music and *guitar* playing). Standard Information Retrieval evaluation measures, such as Precision, Recall or *F*–measure, cannot capture these subtle nuances.

The following example illustrates this issue. Given the song "Wish You Well" by artist `Philipp Frankhauser`. The original (filtered) Last.fm tags are:

```
allboutguitar (100)
allboutguitar.com (100)
blues und rock club Karlsruhe (100)
```

Our autotagging algorithm was able to produce the following list of predicted tags:

```
rock
singer-songwriter
pop
guitar
favorites
classic rock
```

By listening to the song, one can perceive a *blues rock* song with predominant use of guitar, which means that tags such as *rock*, *guitar*, or even *singer–songwriter* can make more sense for the song, from a musicological point of view, than the original tags. Yet, the per–song Precision, Recall and *F*–measure give all a zero score. This might suggest that objective evaluation are not the perfect way to assess the quality of an autotagging algorithm. Future work includes a subjetive evaluation (e.g., through user surveys) of the algorithm's predictions, although it should be considered the impracticabillity for such evaluation, specially for large datasets. Another alternative that can be considered is by using the proposed annotations in a higher level task such as music recommendation (Eck et al., 2008; Zhao et al., 2010). An additional research problem to tackle is on the reliability of social tags. Are tags generated by collaborative effort of a community (the so–called wisdom of the crowds) as consistent as those generated by musicologists? We address this research question in Chapter 4.

The concept of tag similarity can be exploited using contextual information, retrieved from the social data itself. Folksonomies tend to encompass various groups of tags that should reflect the underlying semantic facets of the domain including not only traditional dimensions (e.g., instrumentation), but also more subjective ones (e.g., mood) (Sordo et al., 2010).

Figure 3.20 depicts the tag cloud of the song "Razor Face" by artist *Elton John*. The tags in this case belong to more than one musical concept, thus indicating the richness of this source. Concretely, these tags can be (manually) classified into:

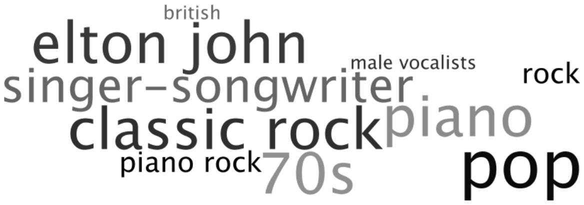

Figure 3.20: Tag cloud of song "Razor Face" by artist *Elton John.*

`Genre` (pop, rock, classic rock), `Locale` (british), `Time period` (70s), `Vocal characteristics` (male vocalists), `Artist name` (elton john), `Artist characteristics` (singer-songwriter), `Instrument` (piano), and even more complex tags which include more than one facet, such as {`Instrument`, `Genre`} (piano rock).

However, the simplicity and user-friendliness of community-based tagging imposes a toll: there is usually no way to *explicitly* relate tags with the corresponding music facets. We address this research problem in Chapter 5, by uncovering the set of semantic facets implicit to the tags of this music folksonomy, and classify tags with respect to these facets. This information, if available, can be used as an additional heuristic for the autotagging algorithm. Learning the meaning of tags is especially helpful for improving the performance of a music autotagger. It can also be extended to solve complementary problems such as tag expansion or ambiguity reduction (Pan et al., 2009).

3.8. Conclusions

In this chapter, we have introduced our proposed automatic music tagging approach. The algorithm predicts tags based on acoustic similarity, using a labeled training dataset. That is, given a seed audio excerpt it propagates tags from previously labeled excerpts that are acoustically similar to the given one, which is opposed to many other approaches that learn models or discriminants from the observations beforehand. This is the case of parametric and semi-parametric methods, such as Gaussian Mixture Models (Turnbull et al., 2008b), Boosting methods ((Bertin-Mahieux et al., 2008)), or Support Vector Machines (Mandel & Ellis, 2008; Ness et al., 2009). In other words, our algorithm propagates tags to an unlabeled audio excerpt, say s, from the k nearest neighbors in the "acoustic space".

Section 3.2 describes our proposed autotagging algorithm. We provided details on the feature extraction, as well as the feature selection process that were followed to reduce the dimension of the audio data, the learning algorithm used, and the parameters that can be tuned to modify the performance of our algorithm.

In order to assess the strength of the proposed autotagging algorithm, we carried out a thorough evaluation of the proposed algorithm in the remaining sections (3.3, 3.4, 3.5, 3.6 and 3.7), using six datasets from different sources. A summary of all the datasets used through this chapter for evaluation is illustrated in Table 3.37.

Table 3.37: Summary of the different datasets used for evaluation.

Dataset	Tag source	#Tracks	#Tags	Categ./Song	Tags/Song
Magnatune-5K (Section 3.3)	Web/ Expert	5481	29	N/A	$\mu = 3.06$ $\sigma = 1.13$
Freesound.org (Section 3.4)	Social tags	260	399	N/A	$\mu = 1.76$ $\sigma = 0.43$
CAL500 (Section 3.5)	Survey	500	174	$\mu = 5.85$ $\sigma = 1.13$	$\mu = 26.04$ $\sigma = 5.74$
MajorMiner (Section 3.6)	Web game	2300	45	$\mu = 2.23$ $\sigma = 0.83$	$\mu = 4.04$ $\sigma = 2.12$
Mood Tags (Section 3.6)	Social tags/ Expert	3469	18	— —	— —
iTMS-500K (Section 3.7)	Social Tags	546386	26324	N/A	$\mu = 5.52$ $\sigma = 5.14$

Our objective in the first experiment (Section 3.3) was to test how the content–based similarity can propagate labels. For styles, we showed that with a 40% annotated collection, we can reach a 78% (40%+38%) annotated collection with a recall greater than 0.4, only using content–based similarity. In the case of moods, with a 30% annotated collection we can automatically propagate up to 65% (30% +35%). These results are quite encouraging as content–based similarity can propagate styles and moods in a surprisingly effective manner. Of course there are limitations for some concepts that have to be clearly encoded in the music for the content–based propagation to work.

As for the experiment with sound effects (Section 3.4), using the Freesound.org database, we first analyzed the tagging behavior of users in the Freesound.org community, where we detected some well–known problems in collaborative tagging, such as polysemy, synonymy, and the scarcity of the existing annotations. Then, we selected a subset of the sounds that are rarely tagged, and proposed a content–based audio similarity to automatically extend these annotations (autotagging). Since the sounds in the test set contained only one or two rare tags, neither precision nor recall were applicable, so we used human assessment to evaluate the results. The reported results show that 77% of the test collection were enhanced using the recommended tags, with a high agreement among the subjects. Although our approach is prone to popular tags, once the sounds are autotagged it allows the users to get a higher recall of those scarcely annotated sounds when doing a keyword–based search.

The aim of the third experiment (Section 3.5) was to first ascertain the parameter estimation for our proposed k–NN autotagging algorithm, and then to present a thorough evaluation of this algorithm for music annotation and retrieval, using the CAL500 dataset (Turnbull et al., 2008b). A qualitative statistical test was performed for a wide range of parameter values, in order to tune the proposed algorithm. Table 3.17 illustrates the best configuration found for both music annotation and retrieval. That is, with $k = 18$, built on top of a PCA reduced feature representation that keeps 75% of the original data variance, using the highlevel descriptors mentioned in Section 3.2.1 and a Euclidean distance measure. Quantitative results on this dataset, however, revealed that using 18–NN for music annotation, although improving per–song evaluation, resulted in a lower tag coverage. In order to tackle this problem, we presented a modification of the algorithm that takes into account the tag modelization from the songs. We call this approach class–based distance classifier (CBDC). Based on centroid–based classifiers from (Han & Karypis, 2000; Kim et al., 2006; Park et al., 2003), this approach computes a centroid (cluster) for each class in the Ground Truth. A song is annotated based on a similarity distance between the song d–dimensional feature vector and the tag clusters. Four different distance measures were tested for the CBDC model, namely Euclidean, cosine, weighted Euclidean and Mahalanobis distance. The latter two distances can be regarded as a single Gaussian modelization of each tag. Experimental results showed that Euclidean distance had a larger tag coverage, and consequently it achieved better per–tag precision and recall. Moreover, additional experiments using both per–song and per–tag evaluation measures revealed that many autotagging models perform well either in per–song or per–tag evaluation, not both. Further evaluation of individual tags, depicted the particularities of each algorithm. For instance, our CBDC model performed comparatively well for emotions that lay on the arousal dimension — emotions such as angry, boring, calm, relax, exciting, etc.— whilst not so well for tags in the valence dimension — specially for happy, sad and their negative examples. These results confirm the findings of Laurier (2011); Yang & Chen (2011) for the specific task of mood classification from audio, where happy songs — which lay on the valence dimension — were the most difficult to classify.

Another purpose of using a naïve model such as CBDC was to emphasize the evidence that a special care must be taken in selecting and capturing a more complete audio–related information, in order to build successful models for automatic tagging of music. In other words, the problem of audio tag classification (and other types of classification as well) cannot be addressed by focusing only on how algorithms learn, but also by understanding what are they learning (Marques et al., 2011).

The fourth experiment (Section 3.6) presented a detailed evaluation of our submission to the MIREX 2011 evaluation contest, for the specific task of Audio Tag Classification. First, the two datasets used for evaluation, namely MajorMiner and MIREX'09 Mood Tags dataset, were introduced. Then, results were reported for binary relevance (classification) and affinity estimation of tags. The evaluation was performed on a 3–fold cross–validation basis, using both per–song and per–tag measures. Our algorithm achieved a good overall rank (between second and fourth) in almost all the experimental results. It is worth mentioning the high recall achieved in the binary relevance (classification) evaluation. We believe that recall is slightly more important than precision in the audio tag classification task. This is explained by the fact that music collections for autotagging are usually weakly labeled. That is to say, the absence of a particular tag annotation for a given excerpt does not necessarily mean that the tag is not relevant to the excerpt. Moreover, proposed results with high recall can be filtered out in a post-processing step using additional information, e.g., contextual information, which can improve precision significantly.

Finally, in the last experiment, our autotagging algorithm was evaluated with a very large dataset, both quantitatively and qualitatively. In order to assess the quality of our results, we compared them against a representative algorithm in the state–of–the–art (Mandel & Ellis, 2008). The quantitative evaluation consisted in capturing the time complexity of each algoritm, by iteratively incrementing the dataset from a small subset of songs to the whole dataset. We monitored the CPU time needed by each algorithm for training and ranking all the tags in the dataset. Results for SVM added more evidence that classic SVM algorithms cannot scale to large data. Our implementation, on the other hand, scaled reasonably well for a half a million song dataset.

Regarding the qualitative evaluation, several evaluations were performed, using both per–tag and per–song evaluation measures. The results showed that our models outperform Mandel & Ellis (2008) in per–song evalations, while the latter performs slightly better in per–tag evaluation. A clooser look at the predicted tags indicated that each model is good at predicting different tags, belonging to the same or to different music facets. In musical genre, Mandel & Ellis (2008) performed releatively well for *house*, *trance*, *drum and bass* or *reggaeton*, where rhythm is more prominent. Our k–NN model, on the other hand, was able to predict a variety of other music genres, ranging from acoustic to electric, from relax to party or aggressive music.

All the datasets, with their corresponding editorial and social metadata, are available online at `http://www.dtic.upf.edu/~msordo/thesis/`.

3.8.1. Contributions

The following is a list of the contributions made in this chapter.

1. An automatic, memory–based, music tagging algorithm that uses acoustic similarity and nearest neighbor classification to propagate labels among songs. The algorithm has the following advantages:

   a) It avoids the design and training of each possible tag, specially for datasets based on folksonomies, where there are thousands of tags.

   b) From the industry perspective, it shows to be scalable in both memory and CPU time consumption, for datasets in the order of tens of thousands of tags and hundreds of thousands of music excerpts (Section 3.7)

2. An exhaustive evaluation of the autotagging algorithm using multiple datasets, for both music and sound effects.

   a) It compares the experimental results with several approaches that are representative of the state of the art of music autotagging (Sections 3.5, 3.6 and 3.7)

   b) It reports and emphasizes the importance and granularity level of different evaluation measures, local or global, for the task of music autotagging (Sections 3.5, 3.6 and 3.7).

   c) Following Aucouturier & Pachet (2004), a set of statistical tests were used for parameter tunning of an audio tag classifier. The outcome of such tests was used throughout Sections 3.5, 3.6 and 3.7 to check the validity of the chosen parameters, and the implications in the achieved performances, both in per–song and per–word evaluations.

   d) Experimental results reveal that a simple model, combined with an audio feature representation that covers a variety of music concepts (including timbre, tonal, temporal and higher level features, such as moods, styles, etc.) can perform as well as, or better than many state of the art approaches. Interestingly, many of the aforementioned state of the art approaches use more complex, time and resource consuming algorithms, though they rely only in timbre information. These results are an additional evidence that a special care must be taken in selecting and capturing a more complete and descriptive audio–related information, in order to build successful models for automatic tagging of music.

   e) In order to add more support to the findings in (b), (c) and (d), another simple autotagging model, called CBDC, was proposed (Section 3.5). The CBDC approach (which stands for Class–Based Distance Classifier) computes a cluster for each class in the Ground Truth. A song is annotated based on a similarity distance between the song feature vector and the tag clusters.

   f) Some additional issues with autotagging were also analyzed, concretely data scarcity, tag correlation (similarity, polysemy) and the impossibility of objective evaluation methods to capture these subtle nuances.

3.8.2. Limitations and future work

In general, the low results in audio tag classification[30] imply that the task at hand is very hard to solve. In this chapter, the results were especially low with a dataset annotated with social tags (Section 3.7). The following subsections highlight the common limitations of current autotagging algorithms, some specific limitations of our proposed algorithm, and finally draw some directions for future work.

Common limitations of autotagging algorithms

On the datasets. Ground Truth datasets for autotagging are usually unbalanced. While some few tags are frequently used, many tags are barely used, which further complicates the problem of learning all the tags in the dataset vocabulary. Datasets are also weakly labeled, the absence of a particular tag annotation for a given excerpt does not necessarily mean that the tag is not relevant to the excerpt. For the same reason, available datasets are often noisy and inconsistent, specially those relying on social tags. Morevover, the nature and source of tags to be used will also depend on the desired application. For example, acoustically objetive tags (Tingle et al., 2010; Torres et al., 2007) can be very useful for high level tasks such as track or artist similarity. Social tags, on the other hand, can be specially valuable for recommending music (e.g., music to listen to while at a party). An open question then is: can we rely on tags obtained from social communities, the so–called wisdom of crowds? We address this problem in Chapter 4.

On the audio features. Most of the state of the art autotagging algorithms follow the bag of frames approach (Aucouturier et al., 2007a). This technique considers the audio signal in a blind way. The audio features are captured on a short-time frame-by-frame basis, using half–overlapping windows of short duration (typically 46ms–50ms). These features are then aggregated to a list or "bag". Finally, a small subset, or the average of this bag of features, is used to train a classifier, using a database of labeled audio excerpts (also known as training dataset). The bag of frames approach has been used in many classification tasks within the MIR field, such as genre (McKay & Fujinaga, 2004; Tzanetakis & Cook, 2002), mood (Laurier, 2011), artist (Mandel & Ellis, 2008) or tag classification (Bertin-Mahieux et al., 2008; Hoffman et al., 2009; Turnbull et al., 2008b; **?**). However, this approach presents some limitations when the problem to be solved is hard (Aucouturier et al., 2007a). For instance, the averaging or the random subsampling process imply the loose of information about temporal dynamics of the audio signal (Aucouturier &

[30] state of the art reports per–song F–measure values ~ 0.5 and per–tag F–measure of ~ 0.3 in CAL500 dataset and MIREX audio tag classification, for instance.

Defreville, 2009; Seyerlehner, 2010). An alternative is to analyze the audio signal block by block (such as the block level features in (Seyerlehner et al., 2010)), thus capturing some local temporal information from the signal itself.

On the evaluation. Several experiments in this thesis have revealed that there is no unique way to assess the quality of autotagging algorithms. Some algorithms perform well in per–song evaluations, however they fail to predict all the tags, especially those that were less frequently used, hence resulting in worse per–tag evaluations. Other algorithms are, a priori, more robust at learning models for all the tags and predict most of the tags, which results in higher per–tag but lower per–song performance. These evaluations are, however, very generic, and do not uncover the particularities of each autotagging system.

On the tag models. Most music autotagging algorithms use the same set of features for training all the tags in the ground truth dataset. Marques et al. (2011) suggest that some tags might benefit from short time audio descriptors, while others only from global features.

On the two stage algorithms. Two stage algorithms use the output of a content–based autotagger as input feature vectors to model each tag in the vocabulary. The rationale behind two–stage algorithms is that they explicitly tackle the problem of tag correlation (Aucouturier et al., 2007b). Results in (Coviello et al., 2011; Ness et al., 2009) show an improvement in classification performance. However the gain of this algorithms in the MIREX evaluation campaign (IMIRSEL, 2011a) is not very significant. In that sense, are we actually reaching a "glass ceiling" (Aucouturier & Pachet, 2004) for audio tag classification, even with novel algorithms that take into account correlations between higher level concepts? Marques et al. (2011) and Coviello et al. (2010) state that some tags might benefit from these two stage algorithms, while some tags won't.

On the tag facets. When using social tags, an autotagging algorithm has no explicit way to relate tags with their corresponding facets (or concepts, or categories). For example, given the tag *Bulgarian hip–hop*, the autotagging algorithm will not have any clue that this tag corresponds to a music genre (hip–hop) from a specific geographic location (Bulgaria). Is it possible to capture this kind of information, in order to help to improve the autotagging algorithm in terms of precision? We address this issue in Chapter 5.

Specific limitations of our proposed autotagging algorithm

On the *k*–NN algorithm. The Weighted vote *k*–NN suffers from popularity bias, specially when *k* is large and/or the voting threshold is more restrictive (see Section 3.2.4 for more details). Tags that are more frequent are more prone to be correctly proposed than less frequent ones. However, the effects of this popularity will depend on the desired application. For example, a weighted vote 18–NN model might predict less tags, but can still be very useful for music search, specially for those audio excerpts that were not annotated or weakly annotated. The 2–NN model (or the CBDC model), on the other hand, might be more valuable for music similarity and recommendation (Eck et al., 2008).

On the dimension reduction. The Principal component analysis (PCA) technique assumes that data is normally distributed (Jolliffe, 2002), but this is not always the case. Empirical results have shown that not all the acoustic features that are used in this thesis follow a Gaussian distribution. Many other audio features ensue, for example, an exponential distribution. Examples of features with a Gaussian distribution include low level features such as *bark bands*, *skewness*, *pitch salience*, *spectral centroid*, *spectral complexity*, *spectral decrease*, *spectral energy*, *spectral spread*. On the other hand, low level features *average loudness*, *dissonance*, *spectral flatness db spectral kurtosis*, and rhythm *first* and *second peak weights* follow an exponential distribution. These distributions were computed using an internal test database. For more information about the features' distributions, please refer to (Freesound.org, 2011). Future work includes using generalized PCA, which can work also for exponential distributions (Collins et al., 2001). We plan to use other alternatives for dimension reduction, for instance Non-negative Matrix Factorization or Relevant Component Analysis (RCA). The latter can help to keep relevant variability only, and to remove irrelevant variability (Shental et al., 2006).

Future work

There are many avenues for future work in music autotagging. Some of these avenues include:

Cross–collection. Traditional research in MIR related Machine Learning tasks consists of using the same Ground Truth dataset for learning models and testing their quality (by performing any validation method). Marques et al. (2011) remark that using a dataset for training and another different dataset for testing negatively influences the evaluation results, and that special care must be taken to understand what are we learning. An additional line of research may include the use of different datasets for training. A high level autotagger can be built by aggregating several autotaggers trained with a different dataset, as it has been recently addressed by Ellis et al. (2011). In addition, each dataset can be defined for a specific music concept (moods, genres, usage, etc.), or alternatively with overlapping concepts which can help to reinforce the classification or misclassification of an audio excerpt. The main drawback of this latter approach is that datasets from different sources tend to share few concepts or use different words for expressing the same concepts. Further work on tag similarity should help overcome these shortcomings.

Evaluation. Future work includes a subjetive evaluation of the algorithm predictions, although it should be considered the current impracticability for such evaluation, especially for large datasets. Another alternative that can be considered is using the proposed annotations in a higher level task such as music recommendation (Eck et al., 2008; Zhao et al., 2010).

Features. Future work includes selecting features that discriminate each concept separately, or uncovering features that are common for a combination of concepts. The latter approach can be helpful for solving the problem of tag correlation. Further work should be devoted also to the way features are aggregated, beyond the classical bag of frames approach (Aucouturier et al., 2007a; Seyerlehner, 2010; Seyerlehner et al., 2010).

Hybrid approach. A multi–faceted approach using expert based classifications, dynamic associations derived from the community driven annotations, and content–based analysis would improve audio tag classification. Some work has already been done for combining such sources of information (Barrington et al., 2009; Knees et al., 2009), with promising results. Future work can be carried out regarding how to effectively combine these diverse sources of information.

CHAPTER 4

Exploring the Semantic Space: Folksonomies and Taxonomies

Never underestimate the power of a million amateurs with keys to the factory.

—Chris Anderson (2006)

4.1. Introduction

Tags are keywords, category names, or metadata that describe web assets and multimedia content. Tags can be selected either from a controlled vocabulary (e.g., a taxonomy), or just by entering a "free text" with no restrictions. When free text tags are introduced by a large community of users to describe the content, they are known as social tags (Lamere, 2008).

For instance, users of the social music website *Last.fm*[1] tagged the artist *Elton John* as "70s", "80s", "pop", "classic-rock", "singer-songwriter", and "british", among others. The combination of all the annotations provided by a community leads to the emergence of a large body of domain-specific knowledge, usually called *folksonomy*.

In Chapter 3, we presented an evaluation of a large music collection annotated with social tags. The obtained results (which achieve state of the art results) were not outstanding —mainly due to the remarkable imbalances, weak tag labelings and inconsistencies in the data. Nevertheless, the experimental results also revealed the limitations of the proposed evaluation measures. Objective measures are not capturing all the subtle details of tag correlations, and also depend on the annotated ground truth. Hence, in this chapter, we try to answer the following research question: can a music autotagging algorithm rely on social tags as a Gold Standard? In other words, can we build data models from music folksonomies that can be useful for annotating music?

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

Folksonomies have an extensive tag coverage, while being updated regularly; whereas taxonomies have a more precise and structured vocabulary. In order to explore this, we analyze whether there is any agreement between a group of music experts and a large community of users, when defining a set of musical concepts, and their relationships. We focus on musical genres and moods, since they are two important aspects when defining music. Also they have been used extensively for the task of music classification, and there exists a lot of expert and social data on these two facets.

4.2. Musical genres

Music genres are connected to emotional, cultural and social aspects, and all of them influence our music understanding. The combination of these factors produce a personal organization of music which is, somehow, the basis for (human) musical genre classification. Indeed, musical genres have different meanings for different people, communities, and countries (Fabbri, 1982).

The use of musical genres has been deeply discussed by the MIR community. A good starting point is the review by McKay (Mckay & Fujinaga, 2006). The authors suggested that musical genres are an inconsistent way to organize music. Yet, musical genres remain a very effective way to describe and tag artists.

Broadly speaking, there are two complementary approaches when defining a set of genre labels: (*i*) the definition of a controlled vocabulary by a group of experts or musicologists, and (*ii*) the collaborative effort of a community (social tagging). The goal of the former approach is the creation of a list of terms, organised in a hierarchy. A hierarchy includes the relationships among the terms; such as hyponymy. The latter method, social tagging, is a less formal bottom–up approach, where the set of terms emerge during the (manual) annotation process. The output of this approach is called folksonomy.

The aim of this section is, then, to study the relationships between these two approaches. Concretely, we want to study whether the controlled vocabulary defined by a group of experts coincide with the tag annotations of a large community.

Section 4.2.1 introduces the pros and cons of expert–based taxonomies and music folksonomies. To compare the similarities between both approaches, we gathered data from two different websites: a musical genre taxonomy from *MP3.com*, and a large dataset of artists' tags gathered from the *last.fm* community. Section 4.2.2 presents these datasets. The experimental results, presented in Section 4.2.3, are conducted in order to analyze the relationships between the genres used in the *MP3.com* taxonomy, and the genre–tags annotated in the artist dataset from *last.fm*. Finally, Section 4.2.4 concludes and summarizes the main findings.

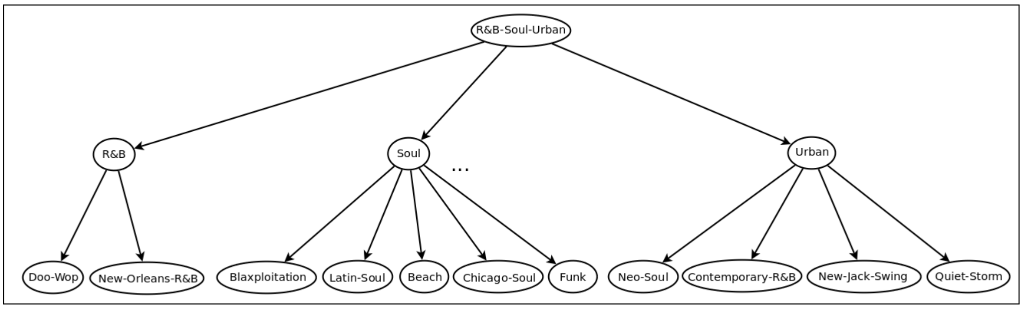

Figure 4.1: Partial view of the *MP3.com* taxonomy, starting with the seed genre *R&B–Soul–Urban*.

4.2.1. Musical genres classification

Expert–based taxonomies

Depending on the application, taxonomies dealing with musical genres can be divided into different groups (Pachet & Cazaly, 2000): Music industry taxonomies, Internet taxonomies, and specific taxonomies.

Music industry taxonomies are created by recording companies and CD stores (e.g., RCA, Fnac, Virgin, etc.). The goal of these taxonomies is to guide the consumer to a specific CD or track in the shop. They usually use four different hierarchical levels: (1) Global music categories, (2) Sub-categories, (3) Artists (usually in alphabetical order), and (4) Album (if available).

Internet taxonomies are also created under commercial criteria. They are slightly different from the music industry taxonomies because of the multiple relationships that can be established between authors, albums, etc. The main property is that music is not exposed in a physical space (shelves). Obviously, exploiting the relationships among the items allows the end–user a richer navigation and personalization of the catalogue.

Furthermore, (Pachet & Cazaly, 2000) shows that there is little consensus among the experts when defining a taxonomy. As an example, using three different musical genre taxonomies (*AllMusicGuide*, *Amazon*, and *MP3.com*) only 70 terms from more than 1500 were common in all the taxonomies.

Music Folksonomies

Since 2004, the explosion of Web 2.0 (e.g., tagging, blogging, user–generated content, etc.) questioned the usefulness of controlled vocabularies (Shirky, 2005). Internet sites with a strong social component, like *Last.fm*, allow users to tag music according to their own criteria. This scenario made the world of taxonomies even more complex.

Nowadays, users can organize their music collection using personal tags like *late night*, *while driving*, etc. As mentioned earlier, new strategies for music

classification have emerged. Folksonomies exploit user–generated classification through a bottom–up approach (Scaringella et al., 2006).
On the one hand, this non-hierarchical approach allows users to organize their music with a better confidence. On the other hand, it creates difficulties for the design and maintenance of expert–based taxonomies, as new terms may emerge from time to time. Thus, in this scenario, up to date expert–based taxonomies become more and more difficult. Yet, it seems reasonable to analyze whether the genres derived from the annotation process share some patterns with the experts' controlled vocabulary.

4.2.2. Datasets

Expert–based taxonomy from *MP3.com*

The *MP3.com* dataset was gathered during September 2005. Table 4.1 shows the relevant information about the genre taxonomy. Experts and musicologists from *MP3.com* identified 749 genres, and organized them in 13 different components, or in other words, the taxonomy has 13 seed-genres. The maximum depth is 6 levels, however, in most of the cases each component has 3 levels (plus the seed–genre at the top). Related with the *staticness* of this vocabulary, as of 2008 it remains the same (only a few more genres were added).

Table 4.1: Dataset gathered during September 2005 from the *MP3.com* expert–based taxonomy. In parenthesis it is shown the number of matched genres in the folksonomy.

Total number of genres	749 (511)
Levels	7
Seed–genres	13
Num. genres at level 1	66
Num. genres at level 2	347
Num. genres at level 3	8
Num. genres at level 4	48
Num. genres at level 5	34
Num. genres at level 6	8

A partial view of the *MP3.com* taxonomy is depicted in Figure 4.1. It shows a component of the taxonomy. The seed genre is *R&B–Soul–Urban*, and the component consists of 3 different levels. A directed edge, e.g., *Urban* ⟶ *New-Jack-Swing*, represents the parent genre (*Urban*) and its subgenre(s), *New-Jack-Swing.*

Folksonomy from the *last.fm* community

A large dataset of artists' tags was gathered from the *last.fm* community during December 2007. Table 4.2 shows some basic information about the dataset.

It is interesting to note that from the average number of tags per artist, 39% correspond to matched genres from the expert taxonomy, whilst the other 61% are distributed among other kinds of tags; including unmatched genres, decades, instruments, moods, locations, etc. For example, the artist *Jade* has the following tags (with their corresponding last.fm normalized weight):

```
Jade: urban(100), rnb(81), 90s(68),
      new jack swing(55), illinois(50),
      r and b(36), ...
```

Table 4.2: Dataset gathered from the *last.fm* community during December, 2007.

Num. artists	137,791
Num. distinct tags	90,078
Avg. tags per artist	11.95
Avg. *MP3.com* genres per artist	4.68
Mapped *MP3.com* genres	511

Nevertheless, since the experiments aim to analyze the agreement between expert genres and genre–tags, we need to match artists' tags with the expert defined genres.

Matching *MP3.com* genres and *last.fm* artist tags. In order to match those tags from the folksonomy that correspond to a genre in the expert taxonomy, a two-step process is followed:

- Compute a normalized form for all the folksonomy tags and expert genres, by:
  - converting them into lowercase,
  - unifying separators to a single common separator
  - treating some special characters, such as "&", which can be expanded to "and" and "n".
- Compute a string matching between the normalized folksonomy tags and expert genres.

The former step is inspired from (Geleijnse et al., 2007). For the latter, a string matching algorithm by Ratcliff and Metzener (Ratcliff & Metzener, 1988) is used to get all possible matches of a tag against a genre from the taxonomy. The algorithm is based on the following measure:

$$sim(s_1, s_2) = \frac{2 \cdot match(s_1, s_2)}{len(s_1) + len(s_2)} \quad (4.1)$$

where *match*(s_1, s_2) represents the sum of the lengths of all the matched substrings between s_1 and s_2. The similarity value goes from 0 to 1. Values close to 0 mean that the two strings are very dissimilar, and a 1 value means that the strings are identical. Deciding which is the threshold for identifying "nearly-identical" words is not trivial. Yet, (Schuth et al., 2007) shows that a threshold of 0.85 gives the best results (the highest *F-measure*).

The following example shows artist *Jade*'s tags that are mapped to an *MP3.com* genre (*90s* and *illinois* tags disappear, and *rnb* and *r and b* are merged):

```
Jade: Urban(100), R&B(100),
      New-Jack-Swing(55)
```

Once the matching process is complete, the next step is to analyze whether the tagging behavior of the community shares any resemblance with the expert taxonomy. The following section presents the experimental results.

4.2.3. Experimental results

In order to measure the agreement between expert genres and the genre–tags defined by the wisdom of crowds, we perform several experiments. Beforehand, we have to compute the "distances" among genres. The next subsection explains the process of computing distances in the expert taxonomy (using the shortest path between two genres), and the tag distances in the folksonomy (by means of a classic Information Retrieval technique, called Latent Semantic Analysis).

The experiments are divided in two main groups. The first set of experiments deal with measuring the agreement at component level (a seed–genre and its subgenres). That is, to validate whether this taxonomy partition (13 components) correspond to the view of the community. Section 4.2.3 present these experiments. The other experiment focuses on the hierarchical structure (levels) of the expert taxonomy. In this experiment the goal is to reconstruct the taxonomy based on the genre distances from the folksonomy (Section 4.2.3).

Computing genre distances

Expert taxonomy. To compute genre distances in the expert taxonomy we simply choose the shortest path between two genres, as an analogy with the number of mouse clicks to reach one genre from a given one (e.g., distance between genres *New–Jack–Swing* and *Soul* is 3, according to Figure 4.1).

Since the taxonomy contains 13 seed genres, a virtual root node is added at the top, thus making the graph fully connected. This way we can compute the path between any genre in the graph.

Folksonomy. Latent Semantic Analysis (LSA), plus cosine similarity, is used as a measure of distance among genres within the folksonomy. LSA assumes a latent semantic structure that lies underneath the randomness of word choice and spelling in "noisy" datasets (Bellegarda, Sept. 2005), such as the one we are using. A significant paper that applies LSA in the music domain is (Levy & Sandler, 2007). The authors show the usefulness of social tags —in a low 10^2 space— to several relevant MIR problems, such as music similarity and mood analysis. LSA makes use of algebraic techniques such as Singular Value Decomposition (SVD) to reduce the dimensionality of the Artist–Genres matrix. After this step, either artist or genre similarity can be computed using a cosine distance. Moreover, Information Retrieval literature (Bellegarda, Sept. 2005; Papadimitriou et al., 1998) states that, after raw data has been mapped into this *latent semantic space*, topic (in our case, genre) separability is improved. For each artist we create a vector based on their (last.fm normalized) genre–tags' frequencies. Once the matrix is decomposed by columns, using SVD with 50 dimensions, we obtain genre similarities. For example, the closest genres to *Heavy Metal* in the semantic space are *Power Metal*, *British Metal* and *Speed Metal*, all with a similarity value above 0.6. On the other hand, similarity between *Heavy Metal* and *Pop* yields a near–zero value.

Agreement between expert and community genres

To measure the agreement between expert defined genre components and community genres we perform two experiments. The first one carries out a coarse–grained similarity (at genre component level), where the main goal is to *separate* the expert genre clusters according to the genre distances in the folksonomy. The second experiment performs a fine–grained similarity (at genre node level) in order to see the correlations between the genre distance in the taxonomy and the distance in the LSA space derived from the folksonomy.

Coarse-grained similarity. The first experiment aims to check how *separable* are the expert defined genre components according to the genre distances in the folksonomy (as defined earlier in Section 4.2.3). The experiment is performed in two steps: (*i*) compute the LSA cosine similarity among all the subgenres within a component (*intra–component* similarity); and (*ii*) compute the LSA cosine similarity among components, using the centroid of each component (*inter–component* similarity).

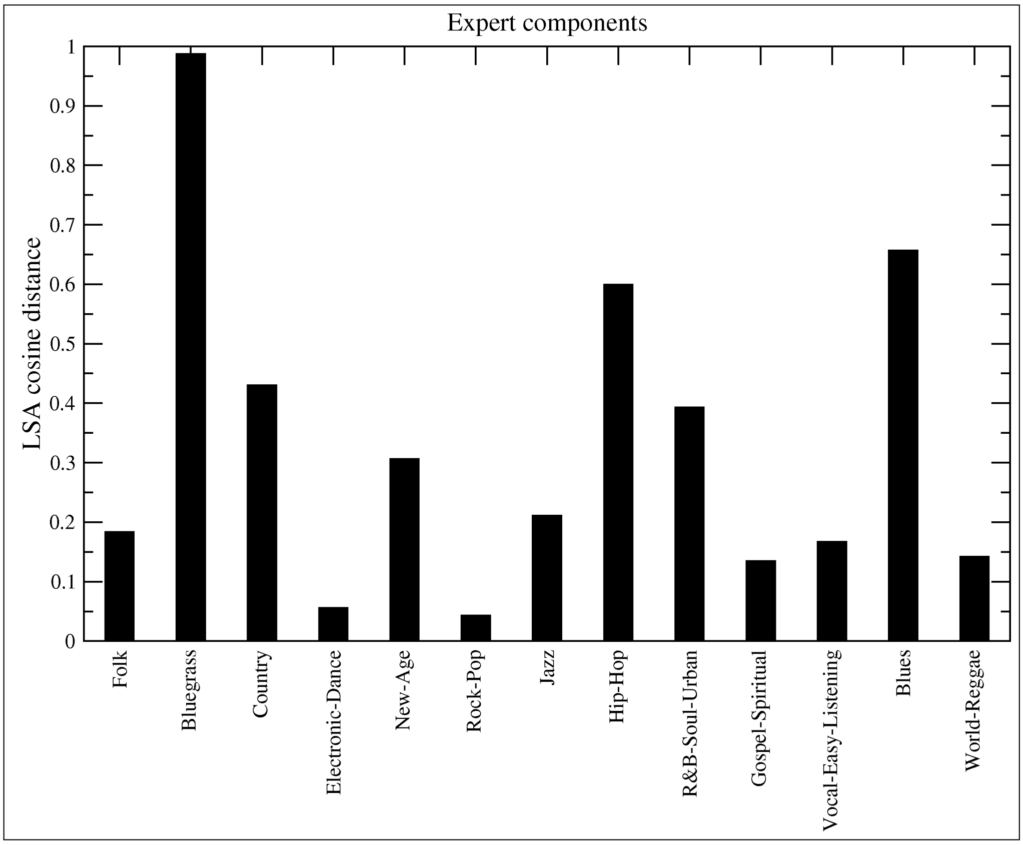

Figure 4.2: Intra–component coarse grained similarity.

The results for intra–component similarity are presented in Figure 4.2. The most correlated components are *Bluegrass*, *Hip-Hop* and *Blues*. Note however that the *Bluegrass* component has only 3 subgenres mapped in our *last.fm* dataset. The components with less community–expert agreement are *Electronic-Dance* and *Rock-Pop*. For the latter genre, it is worth noting that it is an ill–defined seed-genre, and it is also the one including the highest number of subgenres. Some of these *Rock-Pop* subgenres are so eclectic that they could belong to more than one component. For instance, *Obscuro* subgenre is defined in *Allmusic*[2] as "*...a nebulous category that encompasses the weird, the puzzling, the ill-conceived, the unclassifiable, the musical territory you never dreamed existed*".

Regarding the inter–component similarity, we proceed as follows: we compute the centroid vector of each component, and then compare it with the remaining components' centroids. The results are presented in Table 4.3. Note that the results in the diagonal represent the intra–components similarity. For each row, we mark in bold the highest value. Subgenres of *Bluegrass*, *Hip-Hop* and *Blues*,

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

Table 4.3: Confusion matrix for the inter–components coarse grained similarity. For clarification purposes, the diagonal contains the intra–component similarity. The values marked with an asterisk are as significative as the highest value (in bold).

	Folk	**Blueg.**	**Country**	**Elect.- Dance**	**New-Age**	**Rock- Pop**	**Jazz**	**Hip-Hop**	**R&B Soul- Urban**	**Gospel- Spiritual**	**Vocal- Easy- Listening**	**Blues**	**World- Reggae**
Folk	***0.184***	0.144	0.048	0.002	0.040	-0.022	0.007	-0.005	0.003	0.095	0.001	-0.002	0.107
Bluegrass	0.144	***0.987***	0.738*	0.006	-0.018	0.008	-0.019	-0.006	-0.022	0.096	0.014	0.019	-0.033
Country	0.048	**0.738**	*0.430*	0.001	-0.034	0.048	0.007	0.007	0.009	0.054	0.031	0.008	-0.042
Electronic-Dance	0.002	0.006	0.001	*0.056*	0.019	0.012	0.025	0.004	0.019	0.036	**0.171**	0.002	-0.007
New-Age	0.040	-0.018	-0.034	0.019	***0.306***	0.125	0.001	0.022	0.018	0.068	0.102	0.037	0.303
Rock-Pop	-0.022	0.008	0.048	0.012	0.125*	*0.043*	0.036	0.005	0.006	0.040	0.043	0.006	**0.132**
Jazz	0.007	-0.019	0.007	0.025	0.001	0.036	***0.211***	-0.008	0.030	-0.036	0.150	0.046	0.018
Hip-Hop	-0.005	-0.006	0.007	0.004	0.022	0.005	-0.008	***0.599***	-0.005	0.027	0.003	-0.005	0.021
R&B-Soul-Urban	0.003	-0.022	0.009	0.019	0.018	0.006	0.030	-0.005	***0.393***	0.355*	0.009	0.002	0.005
Gospel-Spiritual	0.095	0.096	0.054	0.036	0.068	0.040	-0.036	0.027	**0.355**	*0.134*	0.003	0.163	-0.015
Vocal-Easy-Listening	0.001	0.014	0.031	**0.171**	0.102	0.043	0.150	0.003	0.009	0.003	*0.167*	0.012	0.040
Blues	-0.002	0.019	0.008	0.002	0.037	0.006	0.046	-0.005	0.002	0.163	0.012	***0.657***	-0.004
World-Reggae	0.107	-0.033	-0.042	-0.007	**0.303**	0.132	0.018	0.021	0.005	-0.015	0.040	-0.004	*0.142*

as it has been observed for the intra–component case, are highly correlated in the semantic space. Thus, they are the ones with more agreement between the community and the experts classification. However, only *Hip-Hop* and *Blues* are clearly distinguishable from the rest. On the other hand, according to the community, *Bluegrass* and *Country* genres are very similar. Indeed, other available internet taxonomies, such as *Amazon* or *Allmusic*, include *Bluegrass* as a subgenre of *Country*. Similarly, *Gospel-Spiritual* genre is merged into *R&B-Soul-Urban*.

Fine-grained similarity. In this experiment we focus on the genre level (instead of components). The hypothesis is that genres closer in the semantic space of the folksonomy should also be closer in the expert taxonomy, and vice versa. To validate this formulation a one–way Anova is performed. The independent groups are considered the path distances in the expert taxonomy (ranging from 1..10, the diameter of the taxonomy), whilst the dependent variable is the LSA cosine distance.
Figure 4.3 depicts the box–and–whisker plot. Indeed, a large value of the *F–statistic* as well as a *p–value* smaller than 5% corroborates the hypothesis. Furthermore, to determine the distances that are statistically significant we perform the Tukey's pairwise comparisons test. The results show that path distances 1 and 2 are significant among the rest of the distances, at 95% family–wise confidence level.

Reconstruction of the taxonomy from the folksonomy

In this experiment we try to reconstruct the taxonomy from the folksonomy's inferred semantic structure (LSA cosine distance). We follow a bottom–up approach, starting from the components' leaves. At each step of the process, we record the differences between the inferred and original taxonomies in order to have a similarity metric between them.
The reconstruction of the expert taxonomy from the folksonomy is based on the correct selection of a parent genre, according to the LSA cosine similarity derived from the folksonomy.
The metrics used are: *mean reciprocal rank*, and *root hit.* The *mean reciprocal rank* (MRR) is a statistic widely used in information retrieval. The reciprocal rank (RR) is defined as the inverse of the correct answer's rank, $RR(tag) = 1/rank_{tag}$. For instance, given the *New–Jack–Swing* genre, see Figure 4.4, the closest genre parents (according to the LSA cosine distance) are: (1) *R&B*, (2) ***Urban***, (3) *Traditional–Gospel*, etc. The correct parent genre, *Urban*, is found in the second position, thus $RR(New-Jack-Swing) = \frac{1}{2} = 0.5$. Furthermore we compute whether the top-1 parent belongs to the same component as the child genre (**root hit**). In this example, it is a root hit because both the genre *New–Jack–Swing* and the selected (wrong) parent, *R&B*, belong to the same component, *R&B–Soul–Urban*.

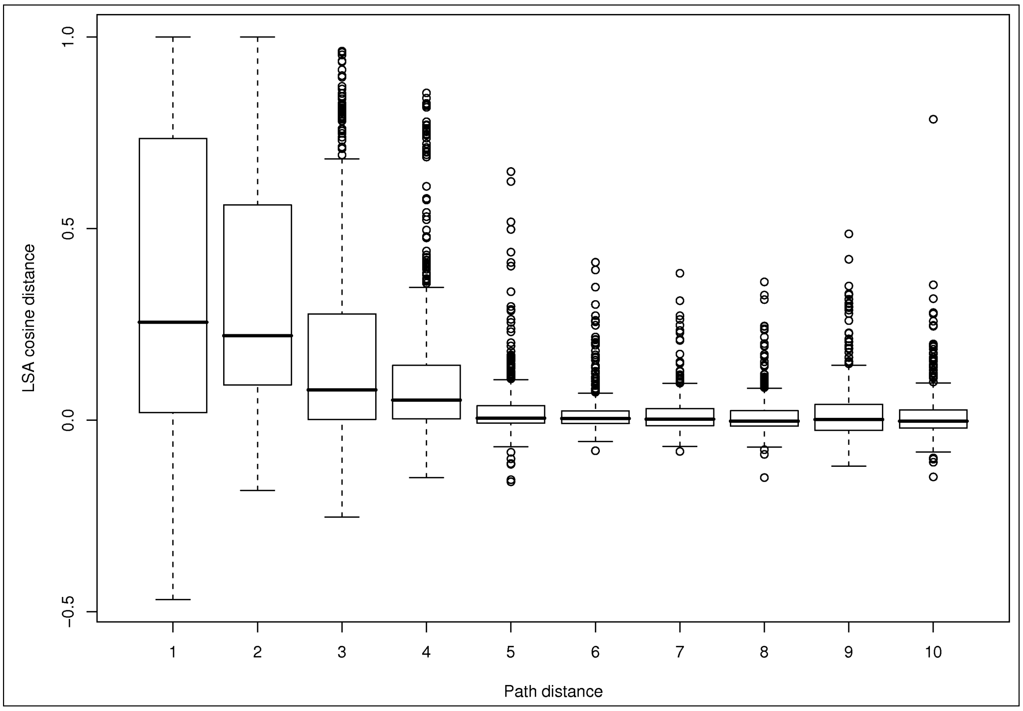

Figure 4.3: Box–and–whisker plot depicting the correlation between genre path distances in the taxonomy and semantic distance from the community (LSA cosine distance). The Anova experiment ($p - value \ll 0.05$) shows that there is a statistical significance among path distances.

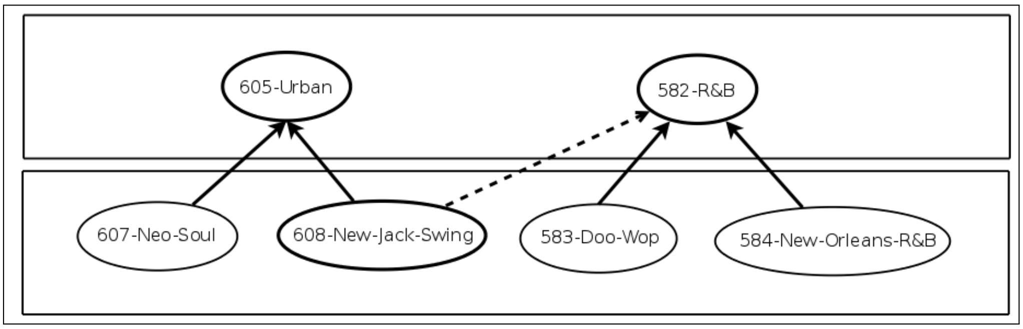

Figure 4.4: Reconstruction of the expert taxonomy from the folksonomy. Selection of parent is made according to the LSA cosine similarity derived from the folksonomy.

Table 4.4 shows the results for each seed–genre. *Bluegrass*' perfect hit rate should be ignored, as the component has only 3 subgenres mapped. The lowest MRR is in *Rock–Pop* genre, which is also the largest (135 genres), and least cohesive component (lowest inter-genre similarity, see Table 4.3). *Hip–Hop*, on the other hand, is a highly cohesive component with a very high MRR. Finally, a lower MRR in *Folk* and *World–Reggae* could be interpreted as a consequence

Table 4.4: Expert matching with Last.fm data.

Seed-genre (component size)	Root Hits (%)	MRR
Folk (22)	36.3	0.447
Bluegrass (3)	100	1.000
Country (35)	85.8	0.636
Electronic-Dance (36)	30.6	0.391
New-Age (5)	40.0	0.700
Rock-Pop (135)	48.2	0.295
Jazz (83)	83.2	0.638
Hip-Hop (22)	81.8	0.894
R&B-Soul-Urban (26)	75.4	0.694
Gospel-Spiritual (7)	38.6	0.558
Vocal-Easy-Listening (11)	27.3	0.446
Blues (43)	81.4	0.455
World-Reggae (83)	53.0	0.389
Weighted Avg. (511)	60.4	0.478

of the taxonomy being too geographically biased. *Bluegrass* and *Country* are seed–genres, even though, as discussed in Section 4.2.3, the folksonomy does not differentiate them clearly; at the same time, disparate genres of all kinds and from all over the world are to be considered sub–genres of *Folk* and *World–Reggae*.

4.2.4. Conclusions

This section presented some interesting findings around musical genres. First of all, the consensus to create a universal taxonomy seems unfeasible. While expert taxonomies are useful for cataloguing and hierarchical browsing, the flat view of folksonomies allows better organization and access of a personal collection.

We presented three different experiments to analyze the agreement between expert–based controlled vocabulary and bottom–up folksonomies. The first two experiments focused on measuring the agreement between genres from the folksonomy and expert genres. A third experiment emphasized the hierarchical structure of a taxonomy, but using information from a folksonomy. In all the experiments the conclusions were the same: some genres are clearly defined both by the experts and the wisdom of crowds, reaching a high agreement between these two views, while it is difficult to get a common consensus of the meaning of other genres.

All in all, experts, wisdom of crowds, and machines[3] agree in the classification

[3] See the results of MIREX'07 in http://www.music-ir.org/mirex/2007/index.php/

and cohesion of some genres (e.g., *Blues*, *Hip-Hop*), and clearly disagree in others (*e.g., Rock*). A musical genre multi–faceted approach using expert based classifications, dynamic associations derived from the community driven annotations, and content–based analysis would improve genre classification, as well as other relevant MIR tasks such as music similarity or music recommendation.

4.3. Moods

Music classification by mood[4] recently emerged as a topic of interest in the Music Information Retrieval (MIR) community. The first task to tackle this problem is to find a relevant representation of mood. In this work, we study mood representations with a bottom–up approach, from a community point of view.

Several works have shown a potential to model mood in music (like (Laurier et al., 2009a; Li & Ogihara, 2003; Yang et al., 2008) , see (Laurier & Herrera, 2009) for an extensive review). Although this task is quite complex, satisfying results can be achieved, especially if we concentrate on the mood expressed by the music rather than the mood induced (Laurier & Herrera, 2009). However, almost every work differs in the way that it represents emotions. Similarly to psychological studies, there is no real agreement on a common model. Comparing these different techniques is a very arduous task. With the objective to evaluate several algorithms within the same framework, MIREX (Music Information Retrieval Evaluation eXchange) (Downie, 2008) organized a task on this topic for the first time in 2007. To do so, it was decided to frame the problem into a classification task with 5 mutually exclusive categories. However, it was shown that these clusters might not be optimal as we suspect some semantic overlap between categories (Hu et al., 2008). In a nutshell, finding the right mood representation is complex.

In this study, we want to address this problem using data collected in an "everyday life" context (not in controlled laboratory settings like in psychological studies). From this data, we want to create a semantic space for mood. In (Sordo et al., 2008), the authors studied the agreement between experts and a community (also based on *last.fm* tags) for genre classification. Levy in (Levy & Sandler, 2007), studied how tags can be used for genre and artist similarity and proposed a visualization of certain words in an emotion space. Both studies inspired our approach of using social tags to compare the semantics of the wisdom of crowds with expert knowledge.

The goal of this section is to create a semantic mood space where we can represent mood and compare it with existing representations. There are two main motivations for this study. First we aim to verify if the knowledge extracted

`Audio_Genre_Classification_Results`.

[4] In order to simplify the terminology, we will use the words emotion and mood independently for the same meaning: a particular feeling characterizing a state of mind.

Table 4.5: Clusters of mood adjectives used in the MIREX Audio Mood Classification task.

Clusters	**Mood Adjectives**
Cluster 1	passionate, rousing, confident, boisterous, rowdy
Cluster 2	rollicking, cheerful, fun, sweet, amiable/good natured
Cluster 3	literate, poignant, wistful, bittersweet, autumnal, brooding
Cluster 4	humorous, silly, campy, quirky, whimsical, witty, wry
Cluster 5	aggressive, fiery, tense/anxious, intense, volatile, visceral

from social tags and the knowledge from the experts (psychologists) converges. Then, we want to generate mood representations that can serve as a basis for further works like music mood classification. In Section 4.3.1, we expose the expert mood representations. In Section 4.3.2, we detail the dataset of tags and then, in Section 4.3.3, its transformation into a semantic space. In Section 4.3.4, we study the categorical representations. In Sections 4.3.4(1) and 4.3.4(2), we generate and analyze dimensional and hierarchical representations. Finally, Section 4.3.5 concludes and summarizes the main findings.

4.3.1. Expert representations

Two main types of representation coexist in the literature. The first one is the categorical model, using for instance basic emotions with around four or five categories including: *happiness*, *sadness*, *fear*, *anger* and *tenderness* (Juslin & Sloboda, 2001). Some works propose mood clusters like the eight clusters from Hevner (Hevner, 1936) (see Figure 4.5) or the five clusters used in the MIREX Audio Mood Classification task, detailed in Table 4.5. The second type of representation is the dimensional model, based originally on Russell's circumplex model of affect (Russell, 1980) (see Figure 4.6). The two dimensions mostly used are arousal and valence[5].

4.3.2. Dataset

Our objective is to obtain a mood space based on social tags. In order to achieve this goal, we need two components: a list of mood words and social network data.

Mood list

For this study, we want to observe the way people use mood words in a social network. We selected words related to emotions based on the main articles in

[5] In psychology, the term valence describes the attractiveness or aversiveness of an event, object or situation.

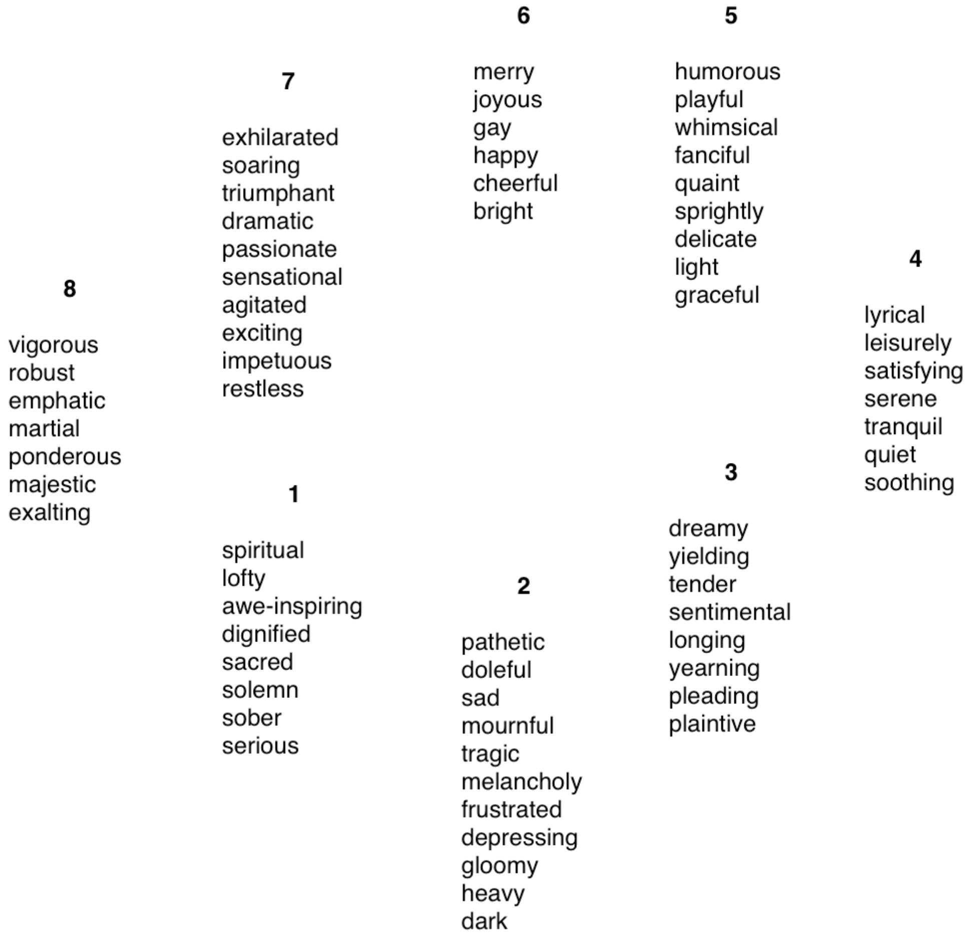

Figure 4.5: Hevner's (Hevner, 1936) model with adjectives grouped into eight clusters.

music and emotion research. We included words from different psychological studies like Hevner (1936) or Russell (1980). We also added words representing basic emotions and other related adjectives (Juslin & Sloboda, 2001). Finally, we aggregated the mood terms mostly used in MIR (Laurier & Herrera, 2009) and the ones selected for the MIREX task (Hu et al., 2008). At the end of this process, we obtained a list of 120 mood words.

Social Tags

Last.fm[6] is a music recommendation website with a large community of users who are very active in associating tags with the music they listen to. With

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

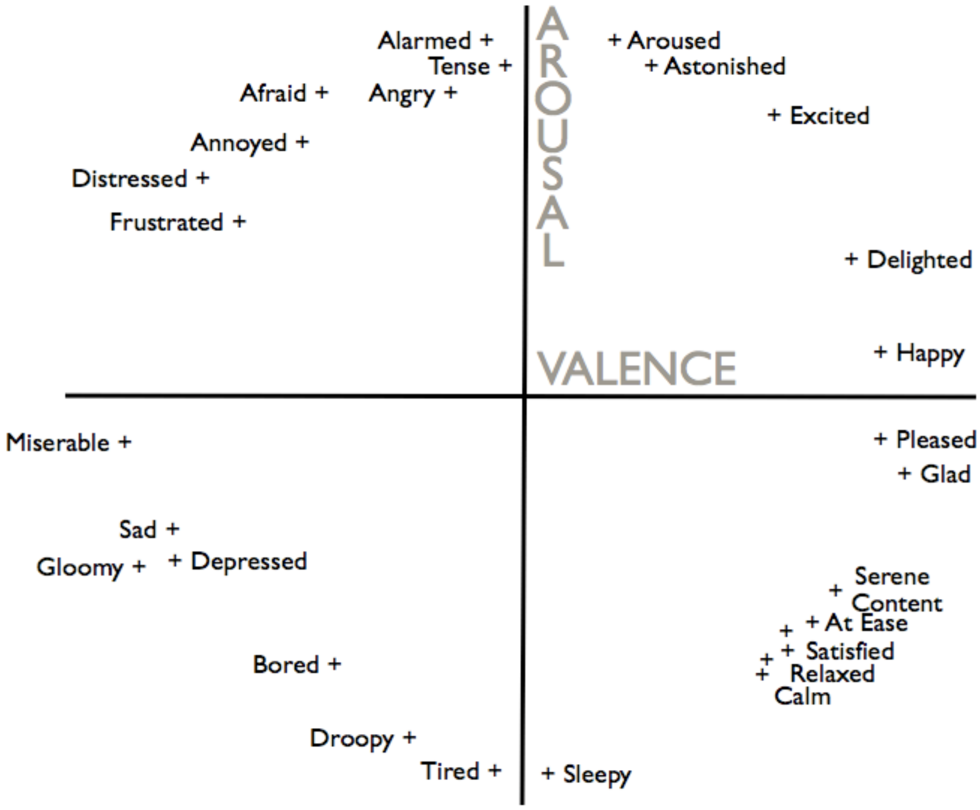

Figure 4.6: Russell's (Russell, 1980) "circumplex model of affect" with arousal and valence dimensions.

over 30 million users in more than 200 countries[7], this social network is a good candidate to study how people tag their music. We crawled 6,814,068 tag annotations from 575,149 tracks in all main genres. From those, 492,634 tags were distinct. This huge dataset contains tags of any kind. From the original 120 mood words, 107 tags were found in our dataset. However some of them did not appear very often. We decided to keep only the tags that appeared at least 100 times, resulting in a list of 80 words. We also chose to keep the tracks where the same mood tag has been used by several users. This subset contains 61,080 tracks. We observe that the mood tags mostly used are *sad*, *fun*, *melancholy* and *happy*. For instance, the tag *sad* has been used 11,898 times in our dataset. On the contrary, the least used tags are *rollicking*, *solemn*, *rowdy* and *tense*, applied in less than 150 tracks. In average, a mood tag is used in 754 tracks.

[7] http://blog.last.fm/2009/03/24/lastfm-radio-announcement

4.3.3. Semantic Mood Space

We aim to compare mood terms by their co-occurences in tracks. Intuitively *happy* should co-occur more often with *fun* or *joy* than with *sad* or *depressed.* This co-occurence information included in the data we crawled from *last.fm* is embodied in a document-term matrix where the columns are track vectors representing tags.

The main problem we have when dealing with this matrix is its high dimensionality and its sparsity (Levy & Sandler, 2009). Consequently, we applied a Latent Semantic Analysis (LSA) (Deerwester et al., 1990) to project the data into a space of a given lower dimensionality, while maintaining a good approximation of the distances between data points. This technique has been shown to be very efficient to capture tag representations for genre and artists similarity (Levy & Sandler, 2007). LSA makes use of algebraic techniques such as Singular Values Decomposition (SVD) to reduce the dimensionality of the matrix. We decided to use a dimension of 100, which seems to be good trade-off for similarity tasks (Levy & Sandler, 2007). In the following experiments, we tried to change this dimension parameter (from 10 to 10 000 on a logarithmic scale), with no significant impact on the outcomes except less relevant results when selecting a too low or too high dimension. Once we have the data into this semantic space, we compute the distance between terms using the cosine distance. The distance values are included in the range $[0, 1]$. Here are some examples of distances between mood tags:

$d_{cos}(happy, sad) = 0.99$
$d_{cos}(cheerful, sleepy) = 0.97$
$d_{cos}(anger, aggressive) = 0.06$
$d_{cos}(calm, relaxed) = 0.03$

We observe that *happy* and *sad* are quite far from each other, as well as *cheerful* and *sleepy.* On the other hand, we note that *anger* is close to *aggressive* and that *calm* is similar to *relaxed.* Even if we show here some prototypical examples, values in the whole distance matrix intuitively make sense.

4.3.4. Experimental Results

Categorical Representations

To study the categorical mood representations, we first derive a folksonomy (community-based taxonomy) representation by means of unsupervised clustering from the social data. Then, we evaluate how the expert taxonomies fit into the semantic mood space.

Folksonomy representation. From our semantic space, we want to infer what would be the ideal categorical representation. To achieve this goal, we apply an unsupervised clustering method using the Expectation maximization (EM) algorithm. This algorithm and the implementation we used (WEKA) are described in (Witten & Frank, 1999). The first important question to be answered is how many clusters should we consider. As we want this number to be inferred by the data itself, we used the *v-fold cross validation* algorithm. We divided the dataset in v folds, training on $v - 1$ folds and testing on the remaining one. We measure the log-likelihood computed for the observations in the testing samples. The results for the v replications are averaged to yield a single measure of the stability of the model. In Figure 4.7, we show the results of this process, displaying an average cost value (in our case 2 times the negative log-likelihood of the cross-validation data). Intuitively the lower is the value, the better is the cluster. To choose the "right" number of clusters, we look at the cost value while increasing the number of clusters. Practically, we stop when the mean cost value stops decreasing and select the current number of clusters.

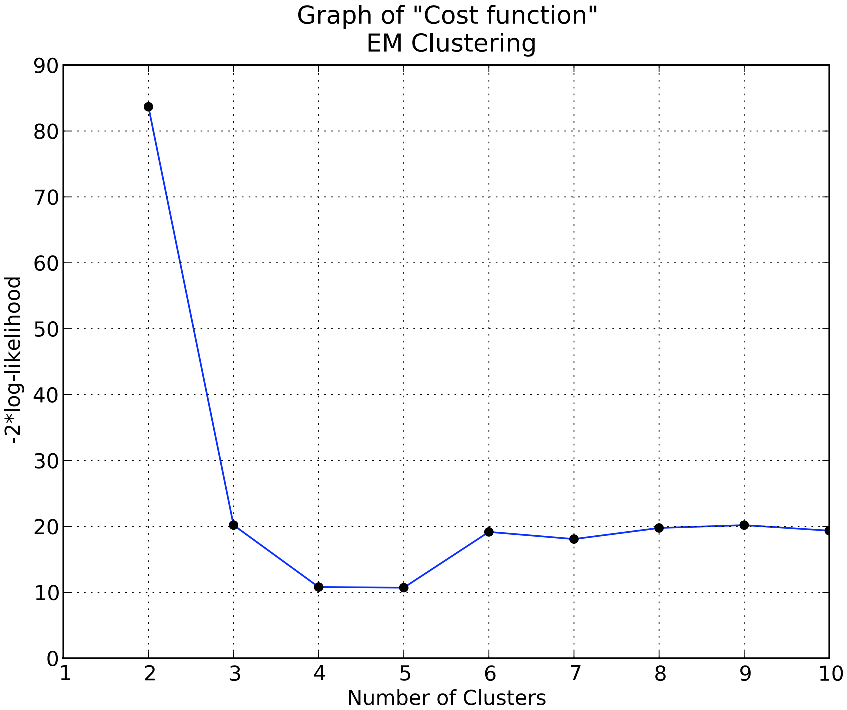

Figure 4.7: Plot of the cost values (2 times the negative log-likelihood) depending on the number of clusters.

We observe that the cost rapidly decreases with the number of clusters until four clusters. After that, it is stable and even increases, meaning that the data is overfitted. Consequently, the optimal number of clusters is four. Using this number for the EM algorithm, we obtained the clusters exposed in Table 4.6.

Table 4.6: Folksonomy representation. Clusters of mood tags obtained with the EM algorithm. For space and clarity reasons, we show only the first tags.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
angry	sad	tender	happy
aggressive	bittersweet	soothing	joyous
visceral	sentimental	sleepy	bright
rousing	tragic	tranquil	cheerful
intense	depressing	good natured	happiness
confident	sadness	quiet	humorous
anger	spooky	calm	gay
exciting	gloomy	serene	amiable
martial	sweet	relax	merry
tense	mysterious	dreamy	rollicking
anxious	mournful	delicate	campy
passionate	poignant	longing	light
quirky	lyrical	spiritual	silly
wry	miserable	wistful	boisterous
fiery	yearning	relaxed	fun

These four clusters are very similar to the categories posed by the main basic emotion theories (Juslin & Sloboda, 2001). Moreover, these clusters represent the four quadrants of the classical arousal-valence plane from Russell previously shown in Figure 4.6:

Cluster 1: angry (high arousal, low valence)
Cluster 2: sad, depressing (low valence, low arousal)
Cluster 3: tender, calm (high valence, low arousal)
Cluster 4: happy (high arousal, high valence)

To summarize, the semantic space we created is relevant and coherent with existing basic emotion approaches. This result is very encouraging and assesses a certain quality of this semantic space. Moreover, it confirms that the community uses mood tags in a way that converges with the basic emotion theory from psychology.

Agreement between experts and community. In this section, we want to measure the agreement between experts and community representations. To do so, we performed a coarse-grained similarity, where we measured how *separable* the expert-defined mood clusters are in our semantic space. First, we computed the LSA cosine similarity among all moods within each cluster (intra-cluster similarity) and then we computed the dissimilarity among clusters, using the centroid of each cluster (inter-cluster dissimilarity). The expert representations we selected for this experiment are the eight clusters from Hevner (see Figure 4.5) where we could match more than 50% of the tags and the five clusters from the MIREX taxonomy (see Table 4.5) where all 31 tags were matched.

Intra-cluster similarity. For each cluster of the expert representations, we compute the mean cosine similarity between each mood tag in the cluster. The results for intra-cluster similarity are presented in Figure 4.8 for the Hevner representation and in Figure 4.9 for the MIREX clusters.

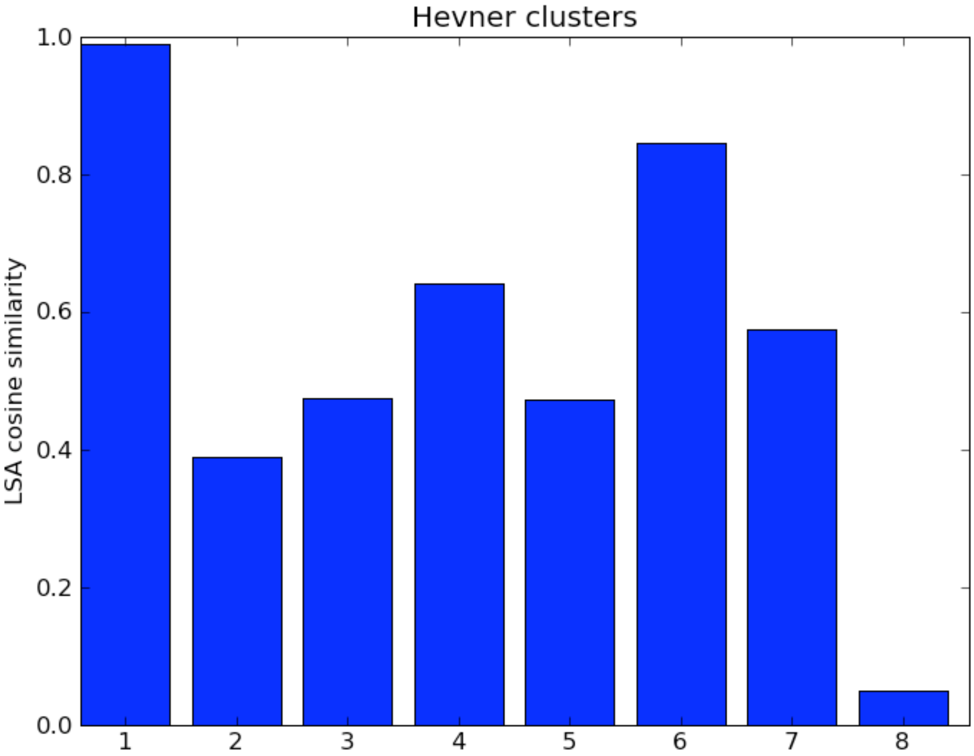

Figure 4.8: Intra-cluster cosine similarity for Hevner's representation.

In the results for the Hevner clusters, we note a high intra-cluster similarity value for cluster 1, which is the one including *spiritual* and *sacred* (please look at Figure 4.5 for the complete list). Cluster 6 performs also quite well (*joyous*, *bright*, *gay*, *cheerful*, *merry*). However we have poor intra-cluster similarity for cluster 8, which includes *vigorous*, *martial* and *majestic*. This might be because these words are also some of the less used in our dataset, but we hypothesize that they are less descriptive today than when the taxonomy was created (1936). Moreover, these words were selected for classical music which is not the main content of the *lasf.fm* music. The rest of the intra-cluster similarity values are in average quite low, meaning that this representation is not optimal in the semantic mood space.

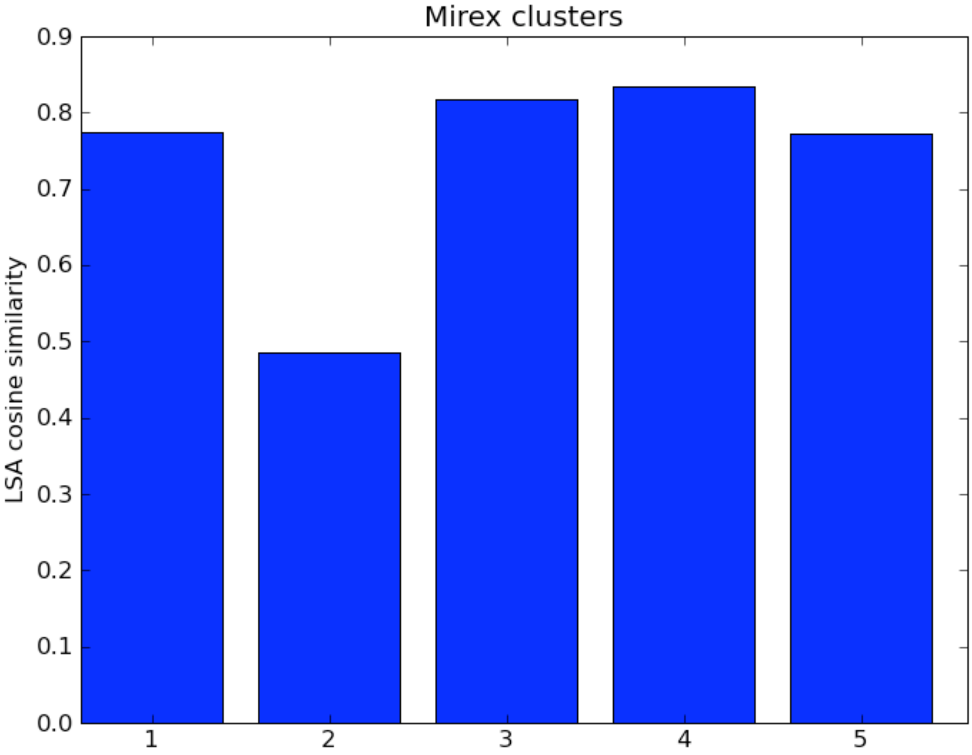

Figure 4.9: Intra-cluster cosine similarity for MIREX representation.

For the MIREX clusters, we remark that the lowest intra-cluster similarity is for cluster 2 (*sweet*, *good natured*, *cheerful*, *rollicking*, *amiable*, *fun*). Maybe is it quite clear that this category is about *happy* music, however the words used are not so common and may lower this value. In average, the intra-cluster similarity value is quite high for this representation. For comparison purpose, we note that the intra-cluster similarity of the folksonomy representation has an average intra-cluster similarity value of 0.82 (see Table 4.8). Obviously, as the folksonomy representation was made from the semantic space itself, it has better results than the other models.

In this part, we have looked at the consistency inside each cluster, however it is also crucial to look at the distances between clusters to evaluate the quality of the clustering representations.

Inter-cluster dissimilarity. To measure how *separable* are the different clusters, we compute the mean cosine distance from each cluster centroid to the other cluster centroids. If we look at our folksonomy representation clusters from Section 4.3.4, the cosine distance between centroids of clusters are all quite high (0.9 in average, see Table 4.8). This is not very suprising as the representation was designed with this data.

Table 4.7: Confusion matrix for the inter-cluster dissimilarity for the MIREX clusters (C1 means cluster 1, C2 cluster 2 and so on). The values marked with an asterisk are the most similar and in bold are the less similar values.

	C1	C2	C3	C4	C5
C1	0	0.74	0.128	0.204	0.108*
C2	0.74	0	0.859	0.816	**0.876**
C3	0.128	0.859	0	0.319	0.265
C4	0.204	0.816	0.319	0	0.526
C5	0.108*	**0.876**	0.265	0.526	0

In Table 4.7, we show the confusion matrix of the inter-cluster dissimilarity for the MIREX clusters. We notice that the lowest value is between cluster 1 and cluster 5, meaning that these clusters are quite similar. This finding correlates with the results from the MIREX task, in which the confusion between these two classes was found significant (Hu et al., 2008). However the confusion between clusters 2 and 4, also relevant in the MIREX results analysis, is not reflected here. Additionally, we observe that the most separated clusters (5 and 2), are also the less confusing in the MIREX results. Looking at the confusion matrix for the Hevner clusters (not shown here for space reasons), we remark that the highest values (dissimilarity above 0.95) are between clusters 7 and 8, and between clusters 1 and 2. On the contrary, the lowest value (0.09) is between clusters 1 and 4. Indeed both clusters have words than can appear similar like *spiritual* and *serene* for instance. We summarize the results of both intra and inter-cluster measures for the different taxonomies in Table 4.8.

Table 4.8: Intra-cluster similarity and inter-cluster dissimilarity means for each mood taxonomy.

Mood Taxonomy	**Intra-cluster similarity**	**Inter-cluster dissimilarity**
Hevner	0.55	0.70
MIREX	0.73	0.56
Folksonomy	0.82	0.9

In a nutshell, the Hevner clusters are less consistent but are more separated than the MIREX ones. Indeed, even if the latter has more intra-cluster similarity, it suffers from confusions between some categories as reflected in our results.

Dimensional representation

Dimensional representation is an important paradigm in emotion studies. To project our semantic mood space into a bi-dimensional space, we used the Self-Organizing Map algorithm (SOM). We decided to use SOM for its topology properties and because it stresses more on the local similarities and distinguishes groups within the data. Because less than half of the Russell's adjectives are present in our dataset, we prefer to compare qualitatively more that quantitatively the expert and the community models. We trained a SOM and mapped each tag onto its best-matching unit in the trained SOM. In Figure 4.10, we plot the resulting organization of mood tags (for clarity reasons, we show here a subset of 58 tags).

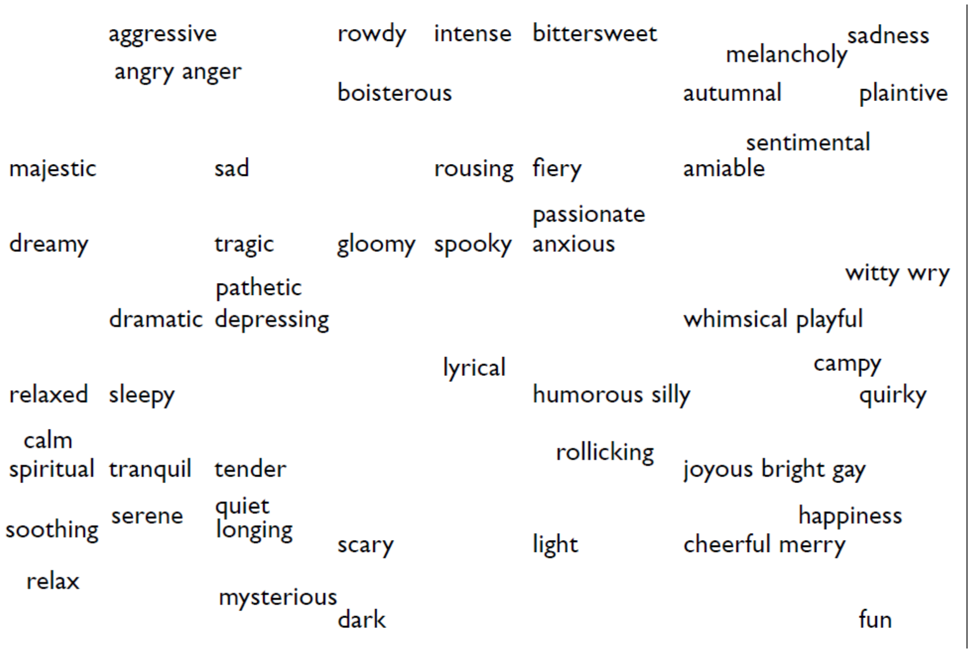

Figure 4.10: Self-Organizing Map of the mood tags in the semantic space.

We observe in the 2D projection four main parts. At the top-left, terms related to *aggressive*, below *calm* and other similar words, at the top-right tags related to *sad* and below words close to *happiness*. We notice the four clusters corresponding to the basic emotions and our folksonomy representation mentioned in Section 4.3.4. This is somehow expected as we already got these clusters from this data. However, having the same results with a second technique confirms our findings. Comparing with Russell's dimensions, we find that the diagonal from top-left to bottom-right is of high arousal. On the contrary, the diagonal from top-right to bottom-left is of low arousal. The vertical axis represents the valence dimension. Even though the 2D representation is not equal, there is a clear correlation between the community and the experts when framing the problem into two dimensions.

Hierarchical representation

The semantic mood space can be visualized in many different ways. In this part we experimented hierarchical clustering techniques to produce a tree diagram (dendrogram). We applied a common agglomerative hierarchical clustering method with a complete linkage (Xu & Wunsch, 2009) and the cosine metric. We used the hcluster[8] implementation. With the 20 most used tags in our dataset, we computed the clustering and plot the resulting dendrogram in Figure 4.11 .

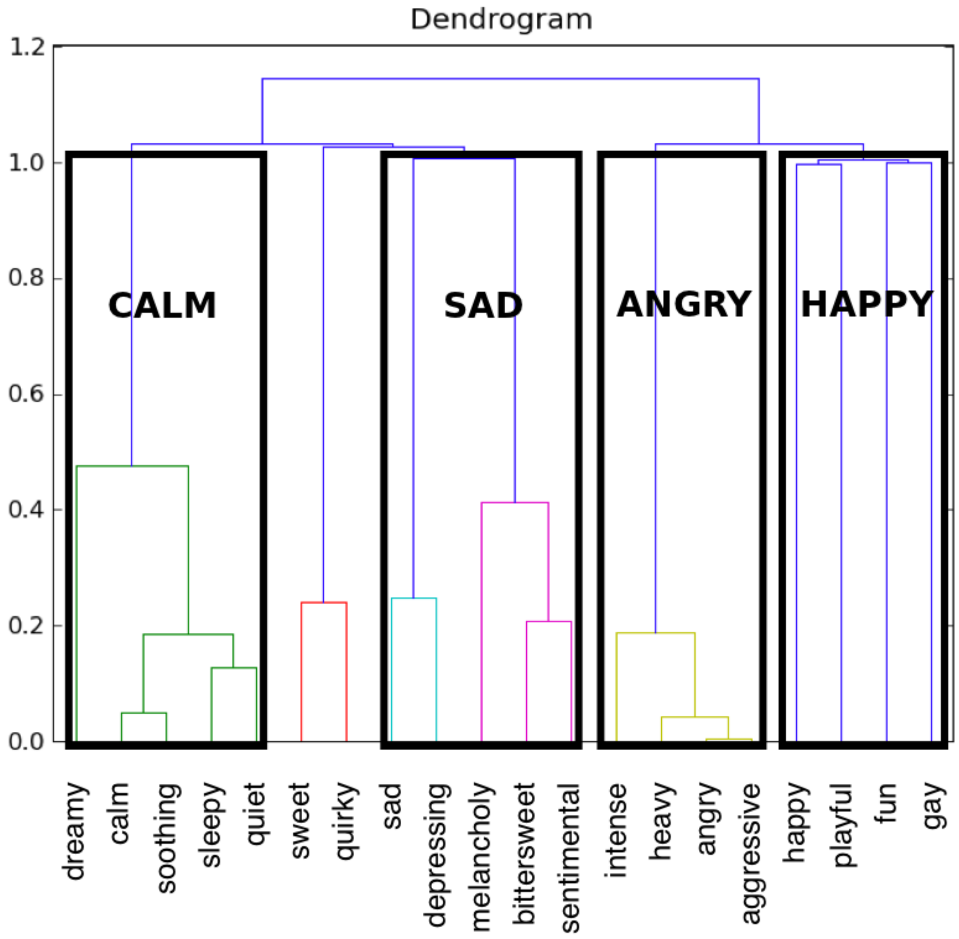

Figure 4.11: Dendrogram of the 20 most used tags.

[8] http://code.google.com/p/scipy-cluster

Although there exists some dendrogram representation of emotions in the psychology literature (Juslin & Sloboda, 2001), the comparison is complex because many of the terms employed are not present in our dataset and also because finding the right metric to measure the similarity between both is not trivial. The hierarchical clustering starts with two branches. Looking at the tags of this first branching, we observe a very clear separation in arousal with *dreamy* and *calm* on the left and *angry* and *happy* on the right. Then the two following branching (resulting in four clusters) represents the four basic emotions also found as the best categorical representation in Section 4.3.4 (from left to right in the dendrogram: *calm*, *sad*, *angry* and *happy*). This confirms another time our findings about the relevancy of these four clusters. Moreover, we notice that the first separation is related to arousal, often considered as the most important dimension. The remaining branches group together similar terms like *angry* and *aggressive* or *sad* and *depressing*.

4.3.5. Conclusions

This last section presented convergent evidence about mood representations. We created a semantic mood space based on a community of users from *last.fm*. We derived different representations from this data and compared them to the expert representations. We demonstrated that the basic emotions: *happy*, *sad*, *angry* and *tender*, are very relevant to the social network. We also found that the arousal and valence dimensions are pertinent. Moreover we have shown that both Hevner's and MIREX representations have advantages and limitations when evaluated in the semantic mood space. The former having better separated clusters and the latter having more consistent clusters. Observations on the confusion and similarity between MIREX clusters confirmed results from previous analysis. We also presented a dendrogram visualization validating again the basic emotion point of view and offering a new representation of the mood space. All these findings show the relevancy of using a mood semantic space derived from social tags. Folksonomy representations can be used in tasks like mood classification or regression to improve the quality of the audio ontent processing algorithms. We can also imagine a visualization of a user emotional states based on his listening habits or history. Moreover, one's musical library can be mapped and explored with a folksonomy representation derived from the whole social network or a particular subset. Finally this approach can be generalized to find other domain-specific representations.

4.4. Summary

In this chapter we studied whether the precise and controlled vocabulary defined by a group of experts correspond with the tag annotations of a large community, the so–called wisdom of the crowds. We ran experiments in two basic musical concepts: music genre and mood, since they are two important aspects when defining music, and they also have been used extensively for the task of music classification. Regarding music genre, the experimental results show a clear agreement for some components of the taxonomy (*Blues*, *Hip-Hop*), whilst in other cases (e.g., *Rock*) there is no correlations. Interestingly enough, the same results are found in the last editions of MIREX results for audio genre classification task. Thus, showing the fact that a musical genre could have a multi–faceted definition. As for moods, we demonstrated that the basic emotions *happy*, *sad*, *angry* and *tender* are very relevant to the social community. With respect to expert–defined mood clustering representations, we found that the arousal and valence dimensions based on Rusell's model of emotion (Russell, 1980) can also be captured. Moreover, we have shown that both Hevner's and MIREX representations have advantages and limitations when evaluated in the semantic mood space. The former having better separated clusters and the latter having more consistent clusters.

This chapter focused on the two facets, music genre and mood, since they are two important aspects when defining music. They have also been used extensively for the task of music classification, and there exist multiple expert representations of genres and moods. Nevertheless, folksonomies cover all possible "ways to talk about music", beyond the musical genres or moods, which are only two facets among many others (e.g., *musical culture*, *record labels*, or *music software*). In Chapter 5, we address this issue. That is, we propose a model to infer the set of semantic facets implicit to the tags of a music folksonomy, and to classify tags with respect to these facets.

CHAPTER 5

Semantic Facets of Music Tags

Knowledge is the conformity of the object and the intellect

—Averroes

5.1. Introduction

Music is a complex phenomenon that can be described according to multiple *facets*. Descriptive facets of music are commonly defined by experts (e.g., stakeholders in the music industry) in professional taxonomies, which typically include all dimensions that can be accounted for in the production and edition of a music piece (e.g., name of artists, recording studio, producer, music genre, etc.). Multifaceted descriptions are especially useful for music browsing and recommendation, as they facilitate non-linear exploration. For instance, recommendations of the Pandora[1] Internet radio use around 400 music attributes grouped in 20 facets (Westergren, 2010),[2] as for instance Roots (e.g., "Afro-Latin Roots"), Instrumentation (e.g., "Mixed Acoustic and Electric Instrumentation"), Recording techniques (e.g., "Vinyl Ambience"), or Influences (e.g., "Brazilian Influences").

However, there exists no consensual taxonomy for music. Previous research showed the music industry uses *inconsistent* taxonomies (Pachet & Cazaly, 2000), even when restricting to a single and widespread facet such as the music genre. Also, expert-defined taxonomies (music-related or not) have two fundamental problems. First, they are very likely to be *incomplete*, since it is impossible for a small group of experts to incorporate in a single structure all the knowledge that is relevant to a specific domain. Second, since domains are constantly evolving, taxonomies tend to become quickly *outdated* —in music, new genres and techniques are constantly emerging. Current lack of consensus

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

[2] http://en.wikipedia.org/wiki/List_of_Music_Genome_Project_attributes

on which are the relevant semantic facets of music, and the inherent inconsistency of some facets (e.g., genre) make the design of a consensual, complete and stable music ontology (including hundreds of facets) a daunting task.

An alternative strategy for describing music consists in relying on the broadness of the web and making use of the "wisdom of the crowds". Many music websites allow users themselves to assign their own descriptive tags to music items (artists, albums, songs, playlists, etc.). For instance, users of the social music website *Last.fm*[3] tagged the artist *Elton John* as "70s", "80s", "pop", "classic-rock", "singer-songwriter", and "british", among others. Their combination of annotations provided by thousands of music users leads to the emergence of a large body of domain-specific knowledge, usually called *folksonomy.* Due to its informal syntax (i.e. direct assignment of tags), the tagging process allows the collective creation of very rich tag descriptions of individual music items.

When compared to taxonomies defined by experts, music folksonomies have several advantages. First, completeness, they ideally encompass all possible "ways to talk about music", including both *lay* and *expert* points of view. Second, due to the continuous nature of the tagging process, folksonomies tend to be well updated. Third, they usually incorporate both *commonly accepted* and *generic* concepts, as well as *very specific* and *local* ones.

It seems reasonable to assume that folksonomies tend to encompass various groups of tags that should reflect the underlying semantic facets of the domain including not only traditional dimensions (e.g., instrumentation), but also more subjective ones (e.g., mood).

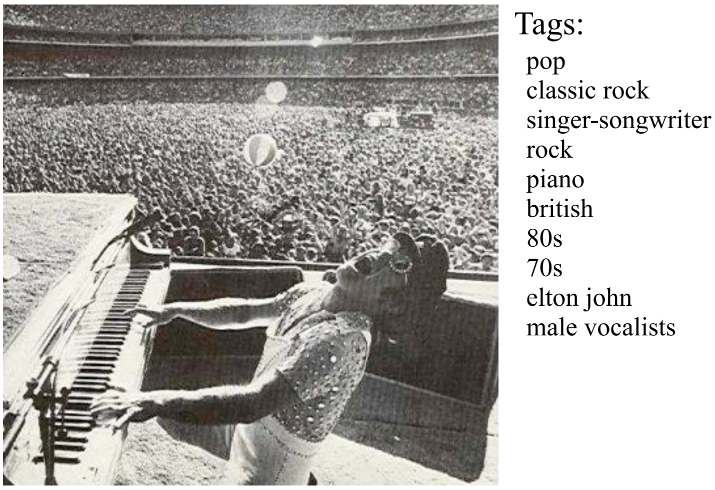

Figure 5.1: List of *Last.fm* tags assigned to *Elton John* artist.

For instance, Figure 5.1 shows a picture of an artist together with a selection of tags attributed by the users of *Last.fm* (the complete list of tags is available on

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

http://www.last.fm/music/Elton+John/+tags). A manual categorization of tags in Figure 5.1 in music facets shows the richness of the tag description of that particular artist: `Genre` (pop, rock, classic rock), `Locale` (british), `Time period` (70s, 80s), `Vocal characteristics` (male vocalists), `Artist characteristics` (singer-songwriter) and `Instrument` (piano).

However, the simplicity and user-friendliness of community-based tagging imposes a toll: there is usually no way to *explicitly* relate tags with the corresponding music facets. When browsing the tag description of a particular artist, *Last.fm* users browse a —albeit very rich— *flat* list of terms. It is for example not explicit in Figure 5.1 that those tags related to music genre are in fact about music genre.

In this chapter, we approach an essential research question that is relevant to bridging this gap: Is it possible to *automatically* infer the semantic facets inherent to a given music folksonomy? A related research question is whether it is then possible to classify instances of that music folksonomy with respect to the inferred semantic facets.

In this chapter, we focus on the music folksonomy obtained from the social music website *Last.fm*. We propose an automatic method for (1) uncovering the set of semantic facets implicit to the tags of this music folksonomy, and (2) classify tags with respect to these facets. We anchor semantic facets on metadata of the semi-structured repository of general knowledge Wikipedia. Our rationale is that as it is dynamically maintained by a large community, Wikipedia should contain *grounded* and *updated* information about relevant facets of music, in practice.

The rest of this chapter is structured as follows: After a review of related work (Section 5.2), we explain in Section 5.3 our approach to obtaining the inherent semantic facets of *Last.fm* tags, and to automatically assigning facets to tags. Results and evaluations are proposed in Section 5.4. We conclude with a summary and directions for future work in Section 5.5.

5.2. Related work

5.2.1. Tag categorization

In (Bischoff et al., 2009) the authors propose an approach, using rule– and model–based methods, to automatically infer the semantic category (facet) of the tags. Rule–based methods rely on regular expressions and predefined lists, whilst the model–based ones employ attributes such as: tag popularity, number of words, number of characters, part of speech, and word sense disambiguation (using WordNet[4]). The list of facets used is: `Topic`, `Time`, `Location`, `Type`, `Author/Owner`, `Opinions/Qualities`, `Usage`, and `Self reference`. Then, they compare automatic tag classification against a ground truth of around

[4] http://wordnet.princeton.edu/

2,100 manually classified tags based on these list of facets. Experimental results using *Last.fm*, *Delicious.com* and *Flickr* datasets show that these two methods can identify tag facets with an accuracy higher than 80%. The results vary significantly across the different domains. Nonetheless, the manual creation of the ground truth—using a handcrafted lists of terms—limits the coverage for these three different domains.

Except the work presented by Bischoff et al. (Bischoff et al., 2009) that includes the music domain in their evaluations, most of the previous work has been focused on the image domain. Overell et al. present in (Overell et al., 2009) a method for classifying Wikipedia articles using structural patterns as features, and WordNet semantic categories as a classification scheme. Then, they apply this method to also classify *Flickr* tags to WordNet semantic categories. Their results show an increase by 115% of the Flickr vocabulary coverage, compared to the WordNet baseline. A similar approach by the same authors is also presented in (Sigurbjörnsson & van Zwol, 2008).

The Semantic Web community has also been working on the problem of inducing an ontology from a corpora derived from social tagging activity (Mani, 2002; Mika, 2005; Schmitz, 2006; Wu et al., 2006).

5.2.2. Social Tagging in Music

Music tags have recently been the object of increasing attention by the research community (Celma, 2010; Lamere, 2008). A number of approaches have been proposed to associate tags to music items (e.g., a particular artist, or a music piece) based on an analysis of audio data (Bertin-Mahieux et al., 2008; Turnbull et al., 2008b), on the knowledge about tag co-occurence (Levy & Sandler, 2008), or on the extraction of tag information from community-edited resources (Sarmento et al., 2009). However, in most cases, such approaches consider tags independently, i.e. not as instances in structured hierarchies of different music facets. When hierarchies of facets are considered, they are usually defined *a priori*, and greatly vary according to authors. For example, (Lamere, 2008) groups tags in the following facets: `Genre`, `Locale`, `Mood`, `Opinion`, `Instrumentation`, `Style`, `Time period`, `Recording label`, `Organizational`, and `Social signaling`.

Alternatively, Pachet et al. (Pachet & Roy, 2009) use 935 labels grouped in 16 facets: `Style`, `Genre`, `Musical setup`, `Main instruments`, `Variant`, `Dynamics`, `Tempo`, `Era/epoch`, `Metric`, `Country`, `Situation`, `Mood`, `Character`, `Language`, `Rhythm` and `Popularity`. Aucouturier (Aucouturier, 2009) considers 801 labels grouped in 18 facets, the same as (Pachet & Roy, 2009), with the exception of `Popularity`, and 3 extra facets: `Affiliate`, `Special creative period`, and `Text category`. On the other hand, Turnbull et al. (Turnbull et al., 2008b) use 135 concepts grouped in only 6 facets: `Instruments`, `Vocal characteristics`, `Genres`, `Emotions`, `Acoustic quality` and `Usage`.

To our knowledge, however, few efforts have been dedicated to the particular task of *automatically* identifying the relevant facets of music tags. In their work on inferring models for genre and artist classification, Levy et al. apply dimensionality reduction techniques to a data set of tagged music tracks in order to obtain their corresponding compact representations in a low-dimensional space (Levy & Sandler, 2008). They base their approach on tag co-occurrence information. Some emerging dimensions can be associated to facets such as `Era` (e.g., the dimension [90s]). However, most of the dimensions thus inferred are, in fact, a combination of diverse music facets, such as for example the dimension [guitar; rock], which includes concepts of `Instrumentation` and of `Genre`, or [seen live; world music] (including concepts of `Social signaling` and `Genre`), or [new wave; 80s] (including concepts of `Genre` and `Time period`).

Cano et al. use the WordNet ontology to automatically describe sound effects (Cano et al., 2004b). Albeit the very large amount of concepts in WordNet, they report that it accounts for relatively few concepts related to sound and music, and propose an extension specific to the domain of sound effects. On the one hand, they illustrate that browsing can indeed be greatly enhanced by providing multifaceted descriptions of items. On the other hand however, it is our belief that, because of their necessary stability, existing ontologies are not the most adapted tool to describe domains of knowledge with inherent open and dynamic semantics, such as music.

5.2.3. Expert-defined music facets

Table 5.1 provides a review of 45 music facets commonly used in the literature. In this review, we refrained from attempting to group together facets with apparent polysemic meanings (e.g., Style, Rhythm) or facets with different denominations yet apparent similar meanings (e.g., Style and Stylings, or Acoustic qualities and Texture). From the list of referenced literature in this table, we specially mention the Music Ontology(Raimond et al., 2010), which is an attempt to provide a vocabulary for linking a wide range music-related information. The ontology is a formal framework for dealing with music-related information on the Semantic Web, including editorial, cultural and acoustic information.

Obtaining detailed taxonomies for all these facets is an elusive task because of two factors: the current lack of consensus in musical taxonomies, and the scarcity of resources (for instance, most references in Table 5.1 do not provide the full list of instances for each facet used nor the full mapping between facets and their instances).

Nevertheless, we gathered specific expert taxonomies for four particular facets, chosen for their relevance in current literature, namely, `Genre`, `Mood`, `Musical Instruments` and `Country and Language`. See Section 5.4.1 for more details.

Table 5.1: Expert-defined music facets and corresponding examples of use in the literature and references. Different meanings of a facet appear in different lines.

Facet	**Example** (as used in the literature)	**References**
Genre	Blues	(Lamere, 2008) (Pachet & Roy, 2009) (Aucouturier, 2009) (Levy & Sandler, 2008) (Turnbull et al., 2008b) (Duan et al., 2008) (Bertin-Mahieux et al., 2008) (Raimond et al., 2010)
Style (1)	BeBop	(Pachet & Roy, 2009) (Aucouturier, 2009)
Style (2)	Political, Humor	(Lamere, 2008) (Bertin-Mahieux et al., 2008)
Leanings/Stylings	Classical Stylings	(Westergren, 2010)
Character	Child-oriented	(Pachet & Roy, 2009) (Aucouturier, 2009)
Roots	Acid jazz roots, Funk roots	(Westergren, 2010)
Influences	Flamenco influences	(Westergren, 2010)
Time period, era/epoch	70s, 1989, Romantic period	(Lamere, 2008) (Pachet & Roy, 2009) (Levy & Sandler, 2008)
Recording label	Kill Rock Stars	(Lamere, 2008)
Locale, country, nationality	Germany	(Lamere, 2008) (Pachet & Roy, 2009) (Levy & Sandler, 2008) (Bertin-Mahieux et al., 2008)
Language	Spanish	(Pachet & Roy, 2009) (Aucouturier, 2009)
Musical setup	String ensemble	(Pachet & Roy, 2009) (Aucouturier, 2009)
Main instrument	Double-bass	(Pachet & Roy, 2009) (Aucouturier, 2009)
Instrumentation , instrument	Piano, female vocal	(Lamere, 2008) (Turnbull et al., 2008b) (Duan et al., 2008) (Bertin-Mahieux et al., 2008)(Westergren, 2010)
Instrumentation	Acoustic rock instrumentation	(Westergren, 2010)
Orchestration, arrangement	n/a	(Raimond et al., 2010)
Performance	n/a	(Raimond et al., 2010)
Vocal characteristics	Agressive, breathy	(Turnbull et al., 2008b)
Vocals (1)	Male, group	(Duan et al., 2008)
Vocals (2)	Breathy, unintelligible vocal delivery	(Westergren, 2010)
Acoustic qualities	Catchy, heavy beat, fast tempo, acoustic texture	(Turnbull et al., 2008b)
Variant	Natural, acoustic	(Pachet & Roy, 2009) (Aucouturier, 2009)
Texture	Acoustic	(Duan et al., 2008)

Table 5.2: Expert-defined music facets and corresponding examples of use in the literature and references. Different meanings of a facet appear in different lines. (Cont.)

Facet	**Example** (as used in the literature)	**References**
Production	Studio, live	(Duan et al., 2008)
Recording techniques	Vinyl ambience, studio production	(Westergren, 2010)
Musical qualities	Easy listening qualities	(Westergren, 2010)
Dynamics	Decreasing	(Pachet & Roy, 2009) (Aucouturier, 2009)
Tempo	Slow, 120 BPM	(Pachet & Roy, 2009) (Aucouturier, 2009) (Turnbull et al., 2008b) (Duan et al., 2008)
Metric	3/4, 4/4	(Pachet & Roy, 2009) (Aucouturier, 2009)
Rhythm (1)	Groovy	(Pachet & Roy, 2009) (Aucouturier, 2009)
Rhythm (2)	Strong	(Duan et al., 2008)
Feel	Driving shuffle feel	(Westergren, 2010)
Tonality	Major, minor	(Duan et al., 2008) (Westergren, 2010)
Structures	Basic Rock Song Structures	(Westergren, 2010)
Organizational, situation	City by night, must check out	(Lamere, 2008) (Pachet & Roy, 2009) (Aucouturier, 2009)
Usage	Waking up, music for making love	(Turnbull et al., 2008b)
Social signaling	Seen live	(Lamere, 2008)
Mood, emotion	Agressive, happy	(Lamere, 2008) (Pachet & Roy, 2009) (Aucouturier, 2009) (Levy & Sandler, 2008) (Turnbull et al., 2008b) (Bertin-Mahieux et al., 2008) (Laurier et al., 2009b)
Affective	Positive, neutral, negative	(Duan et al., 2008)
Arousal	Strong, middle, weak	(Duan et al., 2008)
Opinion, preference	Love, favourite	(Lamere, 2008) (Levy & Sandler, 2008) (Bertin-Mahieux et al., 2008)
Popularity	High, medium, low	(Pachet & Roy, 2009)
Lyrics	n/a	(Raimond et al., 2010)
Lyrical content	Political lyrics, cash-obsessed lyrics	(Westergren, 2010)
Place, festival	Glastonbury	(Raimond et al., 2010)

5.3. Method

Our method consists in using metadata from Wikipedia to infer the semantic facets of the *Last.fm* music folksonomy. This is performed in two steps. In the first step, we specialize the very large network of interlinked Wikipedia pages to the specific domain of the *Last.fm* music folksonomy. This is done by maximizing the overlap between Wikipedia pages and a list of frequent tags from the folksonomy. As the resulting network still represents a very large number of nodes, in a second step, we focus on the most relevant ones (node relevance being defined as an intrinsic property of the network). This step also includes additional refinements.

5.3.1. Data

Our data consists of a large dataset of artist tags gathered from *Last.fm* during April 2010 via the API provided by *Last.fm*.[5] The dataset consists of around 600,000 artists and 416,159 distinct tags. This dataset was cleaned in order to remove noisy/irrelevant data: (1) tags were edited in order to remove special characters such as spaces, etc.; (2) tags were filtered by weight,[6] only tags with a weight ≥ 1 were kept; and (3) tags were filtered by usage, keeping only those tags that were applied ≥ 10 times. As a result, the final dataset consists of 582,502 artists, 39,953 distinct tags, and an average of 9 tags per artist.

5.3.2. Obtaining a music-related network

Wikipedia pages are usually interlinked, and we use the links between two particular types of pages (i.e. *articles* and *categories*) to construct a music-related network. Concretely, we use the DBpedia[7] knowledge base. that provides structured, machine-readable descriptions of the links between Wikipedia pages (DBpedia uses the SKOS vocabulary, in its 2005 version).[8] In particular, we make use of two properties that connect pages in DBpedia: (1) the property *subjectOf*, that connect articles to categories (e.g., the article "Samba" is a *subjectOf* of the category `Dance_music`, and (2), the property *broaderOf*, that connect categories in a hierarchical manner (e.g., the category `Dance` is a *broaderOf* of the category `Dance_music`, which is a *broaderOf* of the category `Ballroom_dance_music`).

We start from the seed category "Music" and explore its neighbourhood from the top down, checking whether connected categories can be considered relevant to the music domain. A category is considered relevant if it satisfies any of the two following conditions:

[5] http://www.last.fm/api

[6] i.e. *Last.fm* "relevance weight", which goes from 0 to 100.

[7] http://dbpedia.org/

[8] http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/

- It is a *Last.fm* tag, such as for example “Rock and Roll”. (This condition will be referred to as *isMusical*);

- At least one of its descendants is a tag from *Last.fm and* the substring “music” is included in the title or the abstract of the corresponding Wikipedia article. (This condition is further referred to as *isTextMusical.*)

The descendants of a category are fetched from DBpedia using the two connecting properties previously described. These descendants can be either “successors” (i.e. all direct *subjectOf* and *broaderOf* of this category), or successors of successors, and so on. This iterative search is limited by a maximum depth, empirically fixed to a value of 4. Indeed, experiments with smaller values yielded a significant reduction of the tag coverage, while experiments with greater values did not increase significantly the coverage.

If any of the previous conditions is satisfied, the category, its successors and their edges are added to the network. Otherwise, the category and all incident edges are removed. The algorithm proceeds iteratively (following a Breadth-First search approach) until no more categories can be visited. A summarized version of the method for obtaining a music-related network is described in Algorithm 5.1.

```
Data: $C = \emptyset$, a list of categories (a queue, initially empty); $N = (V, E)$,
      a directed network with a set of nodes $V$ and a set of edges $E$
      (initially empty);
Result: $N$, network with music nodes;
$C \leftarrow C \cup$ “$Music$”;
while $C \neq \emptyset$ do
    $c \leftarrow first\ element\ of\ C$;
    $C \leftarrow C - c$;
    if (c isMusical) ∨ ((at least one descendant of c isMusical) ∧ (c
    isTextMusical)) then
        $N \begin{cases} V \leftarrow V \cup c \cup successors(c) \\ E \leftarrow E \cup edges\ between\ c\ and\ successors(c) \end{cases}$
        $C \leftarrow C \cup successors(c)$
    else
        $N \begin{cases} V \leftarrow V - c \\ E \leftarrow E - all\ edges\ incident\ in\ c \end{cases}$
    end
end
```

Algorithm 5.1: Pseudo-code for the creation of a network of music-related facets from Wikipedia.

5.3.3. Finding relevant facets

Once the network of music-related facets is built, the next step is to find the nodes that are potentially more relevant to the network than others.

We invert the direction of the edges of the network in order to point back in the direction of the most generic category, i.e. "Music", and we compute the PageRank of the resulting network. PageRank (Page et al., 1999) is a link analysis algorithm that measures the relative relevance of all nodes in a network. In PageRank, each node is able to issue a relevance vote on all nodes to which it points to (thus the need for reorienting the edges). The weight of the vote depends on the relevance of the voting node (i.e. relevant nodes issue more authoritative votes). The process runs iteratively, and (under certain conditions) converges to a stable relative ranking, where nodes to which more edges from other relevant nodes converge (directly or indirectly) are considered more relevant. For initializing the PageRank algorithm, we set the initial weight of each node to 0.

In order to capture general yet complementary facets of music, we aim at reducing semantic overlap as much as possible by applying the following filters:

Stub Filter: We remove all categories with substring "_by_" and "_from_". We noticed that many categories in Wikipedia are actually combinations of two more general categories, as for instance "Musicians_by_genres", which is halfways between "Musicians" and "Music_genres" (see also Figure 5.2). Further, we also remove categories that include "_music(al)_groups" (e.g., "Musical_groups_from_California" that has hundreds of connected categories, hence a high PageRank). Most of these categories are used as *stubs*, even sometimes explicitly so we also excluded categories with the word "stub".

Over-Specialization Filter: We exclude all categories that include lexically a more relevant category. Many relevant categories are *specializations* of other more relevant ones, this occurs mostly with concepts related to anglophone music, which are described in great detail in Wikipedia (e.g., "American_Musicians" includes "Musicians" that has a higher PageRank).

Tag Filter: We remove all categories that are *Last.fm* tags. Our objective is to uncover music facets that are implicit to the tags that make up the folksonomy. In general, tags are *instances* of such facets, not the facets themselves.

5.3.4. Assigning facets to tags

In order to assign a set of facets to a given *Last.fm* tag, we process the subnetwork of Wikipedia pages specialized to the *Last.fm* folksonomy (obtained

in Section 5.3.2), as described in Algorithm 5.2 (Note that this process is restricted to tags that can be matched to one of the nodes in the network).
Given a *Last.fm* tag t, we look at its predecessor categories c, or more formally:

$$predecessors(t) = \{c | (t \in broaderOf(c)) \vee (t \in subjectOf(c))\}$$

If any of these predecessors is a top-N facet (we set $N = 50$), it is then assigned to t. The process continues iteratively until no more facets can be assigned to the tag, or a maximum number of iteration (*maxIter*) is exceeded. This condition can be interpreted as the maximum distance in the network between a tag and a facet.

```
Data: $C = \emptyset$, a list of categories (initially empty); $F$, a list of top-N
      music facets; $t$, a Last.fm tag;
Result: $TF$, list of facets applied to tag $t$;
$iter \leftarrow 1$;
$TF = \emptyset$;
while $(F \neq \emptyset) \vee (iter \leq maxIter)$ do
    $C \leftarrow C \cup predecessors(t)$;
    if $(\exists f \in (F \cap C))$ then
        $TF \leftarrow TF \cup f$
        $F \leftarrow F - f$
    end
    $iter \leftarrow iter + 1$
end
```

Algorithm 5.2: Pseudo-code for assigning Wikipedia facets to *Last.fm* tags.

5.3.5. Tag/Facet relevance

The relevance R_{tf} of a music facet f to a tag t is computed as the normalized inverse distance d_{tf} —in number of successive edges— between t and f:

$$R_{tf} = \frac{d_{tf}^{-1}}{\sum_i \frac{1}{d_{ti}}} \tag{5.1}$$

An example of tag/facet relevance is provided in Figure 5.2.

5.4. Results and Evaluation

After running both stages of our method (sections 5.3.2 and 5.3.3), we obtained a list of 333 candidate facets. Table 5.3 contains the top-50 facets, ordered by pagerank (top to bottom, left to right).
Table 5.4 presents a subset of the obtained facets, followed by a subset of their corresponding list of top tags. Top tags are chosen based on the distance (in number of successive edges in the music network) to the given facet.

Table 5.3: Top-50 Wikipedia music facets.

Music_genres	Aspects_of_music
Music_geography	Hip_hop_genres
Musical_groups	Music_of_California
Music_industry	Music_theory
Musicians	Rock_and_Roll_Hall_of_Fame_inductees
Musical_culture	Musical_subcultures
Occupations_in_music	Recorded_music
Music_people	Musical_quartets
Record_labels	Music_festivals
Music_technology	East_Asian_music
Sociological_genres_of_music	Centuries_in_music
Music_publishing_companies	Musical_composition
Musical_instruments	Musical_quintets
Anglophone_music	Southern_European_music
Music_of_United_States_subdivisions	Music_software
Western_European_music	Incidental_music
American_styles_of_music	Years_in_music
Radio_formats	Music_websites
Music_publishing	Guitars
Albums	Music_competitions
Musical_techniques	Musical_eras
Wiki_music	Music_and_video
Music_history	Musical_terminology
Music_performance	Music_halls_of_fame
Music_publishers_“people”	Dates_in_music

Table 5.4: Random sample of tags inferred for various music facets.

Music_genres	**Occupations_in_music**	**Musical_instruments**	**Aspects_of_music**
Sufi_music	Troubadour	Melodica	Rhythm
Dance_music	Bandleaders	Tambourine	Melody
Indietronica	Pianist	Drums	Harmony
Minimalism	Singer-songwriter	Synthesizers	Percussion
Singer-songwriter	Flautist	Piano	Chords
Music_software	**Music_websites**	**Music_competitions**	**Musical_eras**
Nanoloop	Mikseri.net	Nashville_Star	Baroque_music
Scorewriter	PureVolume	American_Idol	Ancient_music
MIDI	Allmusic	Melodifestivalen	Romantic_music
DrumCore	Jamendo	Star_Search	Medieval_music
Renoise	Netlabels	Eurovision_Song_Contest	Renaissance_music

Figure 5.2 illustrates an example of subnetwork in our data. Given the tag *bulgarian hip-hop*, our method starts navigating through the predecessors of this tag until finally reaching two music facets: *Music_genres* and *Music_geography*. In this particular example, computing the relevance of each facet with Equation 5.1 yields the following:

```
bulgarian hip-hop: {(Music_genres, 0.4),
                    (Music_geography, 0.6)}
```

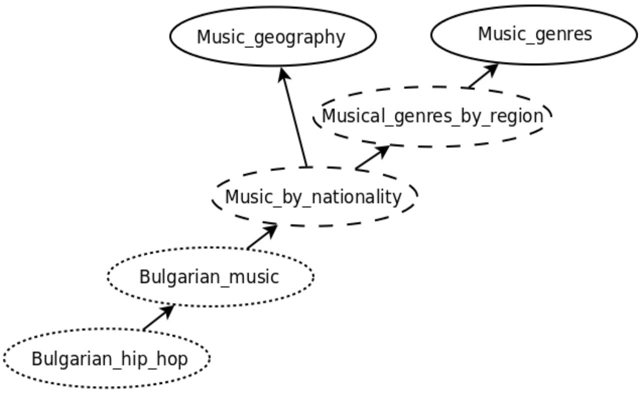

Figure 5.2: Example of subnetwork in our data. Dotted lines correspond to Wikipedia categories that are also *Last.fm* tags. Dashed lines correspond to categories not kept. Plain lines correspond to facets kept.

5.4.1. Evaluation methodology

In this section, we propose to evaluate two aspects of our system. First, in Section 5.4.2, we evaluate the quality of the facets inferred: Is our system able to infer commonly-accepted facets akin to the ones found in taxonomies designed by experts? Is our system able to effectively infer useful facets from the broad, and up-to-date, domain-specific knowledge about music contained in Wikipedia? In order to evaluate the quality of inferred facets, we make use of the expert-defined facets mentioned in Section 5.2.3.

Second, in Section 5.4.3, we evaluate the classification of tags with respect to the inferred facets: Are the tag assignments done by our system similar to what could be found in expert-defined classification systems? In order to evaluate tag classification, we can also make use of the expert-defined facets mentioned in Section 5.2.3, however for this task, we need more than just accepted facets, we also need a ground truth for tags/facets assignments. As mentioned in Section 5.2.3, obtaining detailed taxonomies for all the facets of Table 5.1 is an elusive task. Hence we only focus on four particular facets that are especially relevant in current literature, namely, `Genre`, `Musical Instruments`, `Country and Language`, and `Mood` (see Table 5.1). We call this evaluation dataset our *Gold Standard*, which is detailed in Section 5.4.1.

As put forward in the introduction of this chapter, a central objective to our system is to be able not only to infer music facets and replicate tag/facet classification that could be found in expert systems, but more importantly to provide a *richer* and *more up-to-date* representation of "the words of music." In order to evaluate that particular aspect, we propose to focus part of the evaluation on the comparison of our system with another tag classification system which we call *Baseline System* (detailed in Section 5.4.1). Following the evaluation methodology of (Overell et al., 2009), the Baseline System replicates the be-

havior of our system, but instead of relying on community-based information from Wikipedia, it relies on expert-based information from WordNet.
The two systems are first compared in Section 5.4.3 in terms of tag coverage with respect to the Gold Standard, and also in terms of general quality of tag/facet assignments.
A last evaluation in Section 5.4.4 aims at testing the applicability of our system and at verifying whether community-based information can produce a better *faceted* description of artists' tag sets than expert-based information. In this endeavour, we compare our system with the Baseline system in the task of identifying the facets that exist in the tag description that users have assigned to *Last.fm* artists.

Gold Standard

In this chapter, the Gold Standard (GS) is a list of tags/facets assignments gathered from expert information sources that will serve as ground-truth for evaluation. It is restricted to four particular facets that are especially relevant in current literature on music tags.
Expert-defined instances of the four GS facets have been gathered from Allmusic (http://allmusic.com/) for the `Genre` facet, from references in the psychological literature and taxonomies used in MIREX evaluations (Laurier et al., 2009b) for the `Mood` facet, from MusicBrainz (http://musicbrainz.org/) for the `Musical Instruments` facet and from the United Nations (http://unstats.un.org/), Index Mundi (http://www.indexmundi.com/) and Ethnologue (http://www.ethnologue.com/) for the `Country and Language` facet. There is a total of 1715 tags for the 4 facets in our Gold Standard. Table 5.5 describes the number of tags for each one of these facets, as well as some examples of these tags.

Table 5.5: Number of tags per facet and example of tag/facet assignments in the Gold Standard.

GS Facet	Tags per facet	examples
Genres	711	rock, ambient dub, tango, post-punk, etc.
Instruments	280	bass guitar, trombone, flugelhorn, lute, etc.
Locations & Languages	609	arabic, hungarian, spain, etc.
Moods	115	excited, dark, calm, happy, etc.

Baseline system

The creation of the Baseline System follows the evaluation methodology of (Overell et al., 2009). It is based on a method for network creation and tag categorization that follows a similar overall rationale than the method described in Section 5.3.2 (for the network creation part) and Section 5.3.4 (for the tag categorization part), but it is built on top of expert knowledge from WordNet. Namely, the Baseline System:

1. Uses a third-party knowledge repository (WordNet).
2. Creates a music-related network, in a top-down fashion, starting with initial selected "seeds" from WordNet, and following the links inherent to the knowledge base at hand, adding descendants iteratively to the network.
3. Navigates the network from the bottom to the top, starting from tags of the Last.fm music folksonomy, pruning nodes in the network, and attributing facets to tags.

The main difference between the Baseline system and our proposed method (Section 5.3) are:

1. The use of WordNet instead of Wikipedia as third-party knowledge repository;
2. The connecting properties from WordNet we use are "meronyms" and "hyponyms";
3. The Baseline system does not try to infer facets, they are defined beforehand (expert-defined facets in Table 5.1), and serve as starting seeds to the algorithm (instead of the single "Music" seed in our proposed sytem).

For further details, see Algorithm 5.3. It is important to emphasize the fact that, unlike our system, the Baseline system does not infer facets. It uses expert-defined facets as background knowledge, retrieve corresponding concepts in WordNet and uses WordNet link structure to assign tags to facets. However, not all of the 45 expert-defined facets (Table 5.1) can be matched to WordNet concepts. Only 36 facets were matched. Expert-defined facets that could not be matched to WordNet concepts are: Acoustic qualities, Affective, Dynamics, Lyrical content, Main instrument, Musical qualities, Recording techniques, Social signaling, Variant, and Vocal characteristics.

5.4.2. Evaluating inferred facets

Our system is able to infer 333 facets (Table 5.3 shows the top–50 ones). Unlike for the Baseline system, the facets inferred by our system cannot be exactly matched to corresponding expert-defined facets of Table 5.1. Therefore, we performed a manual match between facets from both sets. In a first stage, we proceed in a strict fashion, only considering matches that can be unequivocally done. This results in matching 8 inferred facets, and are presented in Table 5.6.

```
Data: $C = \emptyset$, a list of categories (a queue, initially empty); $S$, a list of
      seed facets (in the form of WordNet synsets), used as a starting
      point to navigate through the WordNet structure; $N = (V, E)$, a
      directed network with a set of nodes $V$ and a set of edges $E$
      (initially empty);
Result: $N$, network with music nodes;
$C \leftarrow C \cup S$;
while $C \neq \emptyset$ do
    $c \leftarrow first\ element\ of\ C$;
    $C \leftarrow C - c$;
    $N \begin{cases} V \leftarrow V \cup c \cup successors(c) \\ E \leftarrow E \cup edges\ between\ c\ and\ successors(c) \end{cases}$
    $C \leftarrow C \cup successors(c)$
end
while $N$ has Leaves do
    $v = any\ Leaf(N)$;
    if $v$ isNotMusical then
        $N \begin{cases} V \leftarrow V - v \\ E \leftarrow E - all\ edges\ incident\ in\ v \end{cases}$
    end
end
```

Algorithm 5.3: Pseudo-code for the creation of a closed-network of music-related facets from WordNet, using the expert-defined facets as a starting point. The concept of leaf here refers to nodes with $out\_degree = 0$, while $isNotMusical$ is true if the leaf v is not a tag in the folksonomy.

As shown in Table 5.6, our system ranks relevant facets (i.e. expert-defined facets) relatively high. Indeed, seven out of eight (i.e. all except Music_Production) of the expert-defined facets that our system could infer are ranked among the 50 most relevant facets.

Table 5.6: Strict match between inferred and expert-defined facets.

Inferred facet	Expert-defined facet	Inferred facet PageRank
Music_genres	Genre	1
Music_geography	Locale, country, nationality	2
Record_labels	Recording label	9
Musical_instruments	Instrumentation, instrument	13
Music_performance	Performance	24
Music_festivals	Place, festival	34
Musical_eras	Time period, era/epoch	46
Music_production	Production	242

In a second stage, we consider also approximate matches, where one can argue that inferred and expert-defined facets do correspond to similar or related concepts. For a total of 33 additional expert-defined facets, we could produce such "soft" matches. For instance, for some of the inferred facets presented in Table 5.6, we can also obtain the following "soft" matches: `Music_geography` can be matched to `Language`, `Musical_instruments` can be matched to `Main instrument`, or even `Instrumentation`. `Music_production` can also be matched to `Recording techniques`. `Musical_scales` can be matched to `Tonality`. Additionally, we can also find matches between some inferred facets and other, more specific, expert-defined-facets. For instance `Aspects_of_music` can be matched to `Rhythm`, to `Tempo`, to `Metric`, to `Tonality`, and to `Dynamics`. Conversely, some inferred facets are more specific than corresponding expert-defined facets. For example, `Centuries_in_music`, `Years_in_music`, `Musical_eras`, and `Dates_in_music` can be matched to the single expert-defined facet time period, `Era/epoch`.

Globally, i.e. up to rank 242, considering both "soft" and strict matches, we are able to match 41 expert-defined facets, while the Baseline system matches 36 expert-defined facets.

We could also notice cases in which our system was not able to infer correct facets for expert-defined concepts despite the fact that they correspond to existing categories in Wikipedia. For instance, Rhythm, and Lyrics were not inferred because they are also Last.fm tags and our system includes a Tag Filter (see Section 5.3.3).

It is important to notice here that some facets that were inferred do not correspond to any of the expert-defined facets, yet they are potentially useful for music categorization. For instance, the facet "Music_publishing_companies" inferred by our system (see Table 5.3) is clearly relevant (at the very least to actors in the music publishing business), yet is not present in our list of expert-defined facets. Another example are the inferred facets Music_competitions, or Music_websites (see Table 5.4).

There are also examples of inferred facets which are difficult to evaluate when comparing to expert-defined taxonomies. For example, our system infers the facet **Guitars**, to which are assigned *Last.fm* tags such as Bass_guitars, Portuguese_guitar, John_Frusciante, or The_Beatles. Clearly, there is no match with any expert-defined facet. However, one could argue that expert-defined taxonomies focus on describing the qualities of the music itself, and leave aside other parts of the universe of music, that are in fact meaningful to (at least some) users. Indeed associating The Beatles to a "guitar-type music" does make sense.

Another important aspect of our system is the specifity of the inferred tags, such as *iPhone*, *American Idol*, *Jamendo*, *Garage Band*, or even artist's names, which could not be found in the Baseline system. For instance, our system proposes the following description for tag *iPhone*:

```
IPhone: {(Music_software, 0.625),
                          (Music_technology, 0.375)}
```

These particular examples can show the importance of using a community-edited knowledge repository, in terms of being more up-to-date. Nevertheless, there is no way to objectively evaluate such new and unseen tags (instances in classification). This task can only be done by performing a human evaluation of these inferred tags (see Section 5.5).
Looking at the list of top facets generated by our method, we can also observe that there is a certain bias towards anglophone music. For example among the top 50 tags, there are four facets that are clearly related with Anglophone culture –"Anglophone Music", "Music of United States subdivisions" , "American styles of music" or even "Music of California" – while there are only two facets explicitly related with other parts of the world – "East Asian Music" and "Southern European Music". We believe that this is a direct consequence of several factors. First, despite the fact that Last.fm's music repository is open to many different cultures and sub-cultures, musical content and corresponding tag assignment is probably biased towards anglophone music. Second, the Last.fm's folksonomy itself is mostly in English. Third, the english-language version of the Wikipedia used by our methods has obviously an intrinsic bias towards anglophone cultures.

5.4.3. Evaluating assignment of facets to tags

Another important aspect that we want to evaluate is the assignment of (inferred) facets to tags: we wish to evaluate whether the different facets we infer are correctly related with tags. For that we focus on the four facets that are part of our Gold Standard (see Table 5.5). For each of these four reference facets, we compare the tag/facet assignments of our system with the ones obtained by the WordNet-based Baseline System.
We consider two different scenarios. In scenario $\mathcal{S}_1$, we focus on Precision and Recall, we compute the overlap between *all* tags assigned by each system to each Gold Standard facet, and the actual ground-truth tags in those Gold Standard facets. In scenario $\mathcal{S}_2$, we focus on Precision@N.

Precision and Recall

Figure 5.3 depicts the Precision/Recall curve of our system assignments of tags to the facets `Music_genres` (`Genre` in the GS), `Music_geography` (`Locale, country, nationality` in GS) and `Musical_instruments` (`Instrument` in GS). For the `Genre` and `Instrument` facets, it seems that threshold of 4 (`maxIter` in Algorithm 5.2, i.e. the maximum distance from the tag to the candidate facet) gives the best trade off between precision and recall (largest AUC, or Area Under the Curve), whilst this value is reduced to 2 in the case

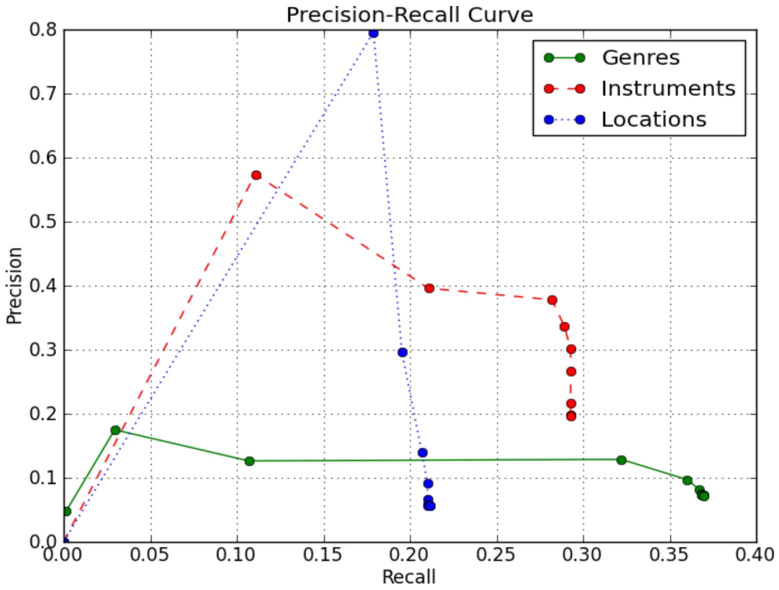

Figure 5.3: Precision/Recall curve of the assignment of facets (`Genre`, `Instrument` and `Location`) to tags, when we vary the threshold for attribution of facets to tags in our system.

of `Locale, country, nationality`. This particular case gives a good insight of our system. In fact, when we impose a threshold of 2, our method is able to assign the facet `Music_geography` to 140 tags, where 79% are actually tags in the GS facet `Locale, country, nationality`. When the threshold is increased to 3, Precision drops dramatically to 29% (165 out of 559). However, by manually looking at these false positives, one can see that effectively most of them are music genres (*Rai*, *Merengue*), artists (due to the nature of the last.fm tags used in this work[9]) or the combination of two facets, such as, e.g., *Spanish_rock* or *Finnish_hip_hop*. For instance, given the tag *Rai*, which the GS describes as a Genre, our system, based on Algorithm 5.2 and Equation 5.1, and using the whole list inferred facets (333), is able to produce the following description:

```
Rai: {(Algerian_styles_of_music, 0.67),
      (Music_genres, 0.16),
      (Music_geography, 0.16)}
```

which is a more complete semantic description of the tag *Rai*. In this case, it seems also clear that a human evaluation (either expert or lay) is needed for evaluating these complex semantics.

Setting our system's threshold to 4 (value chosen after evaluating its performance using the Precision/Recall curve, see Figure 5.3), we compare our

[9] people tend to tag an artist with the name of the artist.

system to the Baseline System in terms of Recall (in Table 5.7) and precision (in Table 5.8).

Table 5.7: Evaluation scenario $\mathcal{S}_1$: Overlap with the Gold Standard ground truth tags in number of tags (Recall value in parenthesis).

GS Facet (total number of tags)	Baseline system classification	Our System classification
Genres (711)	32 (R=0.045)	272 **(R=0.382)**
Instruments (280)	67 (R=0.239)	79 **(R=0.282)**
Locations & Languages (609)	89 (R=0.146)	133 **(R=0.218)**
Moods (115)	3 **(R=0.026)**	0 (R=0)

Table 5.8: Evaluation scenario $\mathcal{S}_1$: Number of tags assigned to each facet (Precision value in parenthesis).

GS Facet	**Baseline system classification**	**Our System classification**
Genres	103**(P=0.311)**	1757 (P=0.155)
Instruments	200 (P=0.335)	208 **(P=0.380)**
Locations & Languages	248 **(P=0.358)**	1207 (P=0.110)
Moods	**51 (P=0.058)**	0 (P=0)

An interesting result of Table 5.7 is the fact that our system more than doubles the coverage of Gold Standard tags of the Baseline system: where the Baseline system covers 191 tags out of the 1715 tags of the Gold Standard, we cover 484, this represents an increase of 153%.

On the negative side however, our system is not able to infer tags from the `Mood` facet. In fact, this facet was not discovered by our method, starting from the seed `Music` and following our algorithm, our system is unable to reach the facet `Mood`, although the concept does exist as a category named `Emotion` in Wikipedia, but our algorithm could not uncover the path in Wikipedia between `Music` to `Emotion`.

We can also notice that our system scores worse than the Baseline system in terms of precision (see Table 5.8), except for the case of the `Instruments` facets. This is due to the fact that our system tends to be much more productive (i.e. it classifies more tags) than the Baseline System. We propose to scrutinize this aspect further by considering an additional evaluation scenario, $\mathcal{S}_2$, where the metric is Precision@N.

Precision@N

As our system tends to be much more productive than the Baseline system, one could argue that precision values would not be directly comparable because of different numbers of tags retrieved by the two systems. Therefore, the objective of evaluation scenario $\mathcal{S}_2$ is to compare precision of the two systems on a comparable base of retrieved tags. For a given facet, we limit the number of tags used in computing precision to the number of tags retrieved by the less productive system, which is always the Baseline system.[10] For instance, for the Genre facet, we compute the precision with respect to the GS ground truth over a set of 103 tags. In the case of the Baseline system, all tags retrieved are used, while in the case of our system, a selection of 103 tags among the 1784 tags retrieved are used.

In order to select among the tags retrieved by our system those that will be used for computing precision, we retrieve the top–N tags (hence, *Precision@N*) for each facet (where N is the limit number). For each facet, the tags are ordered by relevance. This relevance is measured as the distance between the facet to the candidate tag (i.e. the lower this value, the more relevant the tag is).

Table 5.9: Evaluation scenario $\mathcal{S}_2$: Precision@N, where N varies with the facets.

GS Facet (N)	**Baseline system classification**	**Our System classification**
Genres (103)	32 (**$P_{@103}$=0.311**)	27 ($P_{@103}$=0.262)
Instruments (200)	67 ($P_{@200}$=0.335)	76 (**$P_{@200}$=0.38**)
Locations & Languages (248)	89 ($P_{@248}$=0.358)	118 (**$P_{@248}$=0.476**)

It is interesting to compare the results of Table 5.8 to those of this scenario, described in Table 5.9. Logically, the performance of the Baseline system are the same in both tables, but our system appears to perform better with the Precision@N metrics than with Precision.

One can see in Table 5.9 that our system outperforms the Baseline system on the `Instruments` and `Locations` and `Languages` facets. The Baseline system still shows higher precision on the `Genres` facet, but in a much smaller measure than in Table 5.8.

For the case of `Genres`, it should be noted that relatively low precision does not necessarily mean that retrieved items are bad. For instance, it is informative to consider examples among the 76 (i.e. $103 - 27 = 76$) top ranked genres that our system retrieves and that are *not part of the Gold Standard*, as well as the remaining, supposedly not relevant, 71 genres retrieved by the Baseline system that were not part of the Gold Standard either (Tables 9 and 10 in the Appendix provide lists of these genres). One can observe that, even if not part of the Gold Standard, many of the tags retrieved and classified as genres are actually relevant music genres. Both systems were able to retrieve what appears to be very specific genres, such as *Cool_Jazz*, *Neo_Jazz* and

[10] Except in the case of Mood for which our system does not retrieve any tag (see above). For this reason, we will not consider the Mood facet in evaluation scenario $\mathcal{S}_2$.

Psychedelicrock (from the Baseline system); or *Zarzuela*, *Levenslied*, *Schlager*, *Rautalanka* (from our system). This illustrates the reported fact (Pachet & Cazaly, 2000) that expert-based taxonomies of musical genres such as the Gold Standard used here tend to be *incomplete* and *outdated*, albeit comprising over 700 genres.

Moreover, an advantage of our system over the Baseline is to be geared towards retrieving tags that do not necessarily exist in expert-based taxonomies, while the Baseline is biased to be conservative and to retrieve tags corresponding to expert knowledge (either from the Gold Standard, or from WordNet). Therefore, even if the Baseline appears to have a high precision, it is precisely biased towards high precision, the downside of this being its relatively low recall (much lower than that of our system, see Table 5.7). Further, most of the 71 genres from the Baseline system that are not part of the Gold Standard but that are nevertheless relevant are also inferred by our system, beyond the top 103, e.g., *Zydeco*, *Progressiverock*, *Punkrock*, etc. Finally, looking at examples of genres retrieved by our system (see Appendix C) also shows tags not present neither in the Gold Standard, nor in the set retrieved by the Baseline, but that are still relevant, such as e.g., *Indietronica* or *Ethereal_ wave* (in the top 103), and e.g., *Mexican_ Cumbia*, *UK_ Hard_ House* (beyond the top 103). The fact that our system can retrieve such genres is interesting especially because they correspond to a very specialized (either culturally or temporally) corner of human knowledge about music, hardly reachable for small groups of experts, but at the reach of folksonomies.

5.4.4. Use case: assigning semantic facets to artist tags

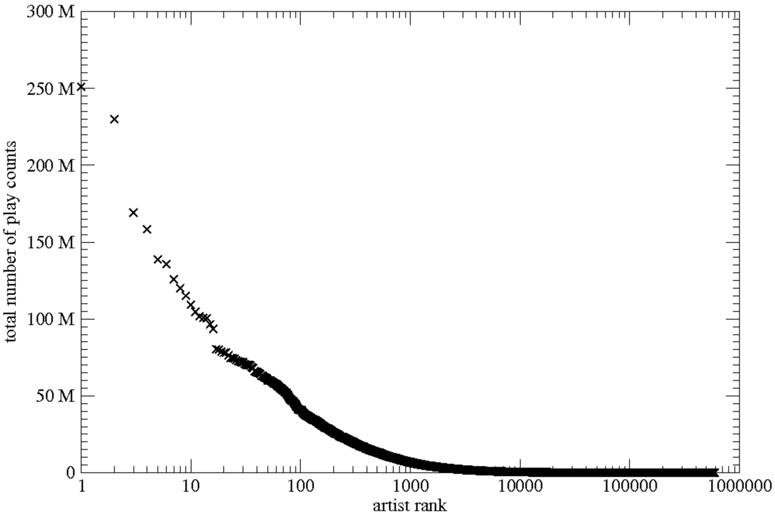

Figure 5.4: The Long Tail for artist popularity. A log–linear plot depicting the total number of plays. Data gathered during April, 2010, for a list of 594,989 artists.

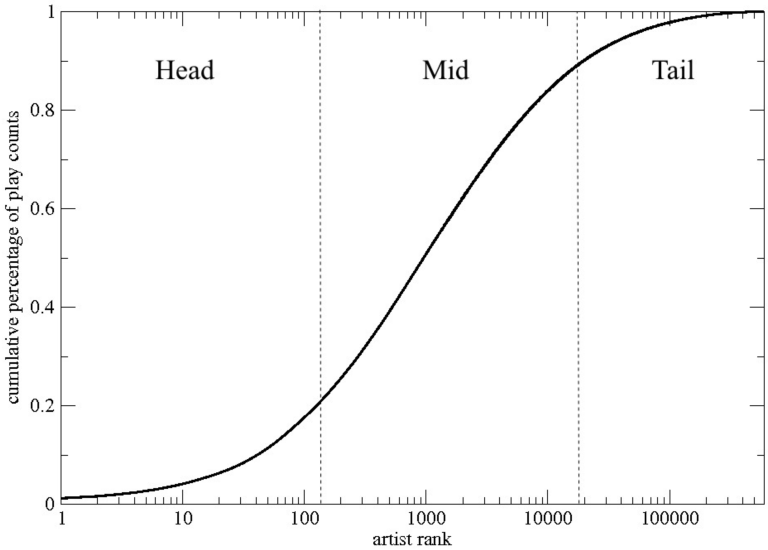

Figure 5.5: The Long Tail model. It shows the cumulative percentage of playcounts of the 594,989 music artists from Figure 5.4. Only top–955 artists, 0.16% of all the artists, accumulates the 50% of total playcounts (N50). Also, the curve is divided in three parts: head, mid and tail ($X_{head \to mid} = 104$, and $X_{mid \to tail} = 10,860$), so each artist is located in one section of the curve.

Here, we wish to verify whether we can produce a meaningful *faceted* description of music artists' tag sets. We are particularly interested in evaluating the worth of community-based information (i.e. Wikipedia) in generating applicable facets and tag/facet assignments for music artists in the *Last.fm* community. Furthermore, we focus on artist popularity (number of total playcounts in *Last.fm*) in order to understand the intrinsic differences that exist in any social tagging system. That is, users tend to tag more the popular artists rather than those that are less known.

Artist Popularity

Figure 5.4 depicts the Long Tail popularity, using total playcounts, for 594,989 artists. The horizontal axis contains the list of artists ranked by its total playcounts. E.g. *The Beatles*, at position 1, has more than 250 million playcounts. The Long Tail model, $F(x)$, simulates any heavy–tailed distribution (Kilkki, 2007). It models the cumulative distribution of the Long Tail data. $F(x)$ represents the share (%) of total volume covered by objects up to rank x:

$$F(x) = \frac{\beta}{(\frac{N_{50}}{x})^{\alpha} + 1} \tag{5.2}$$

where α is the factor that defines the S–shape of the function, β is the total volume share (and also describes the amount of latent demand), and N_{50}, the median, is the number of objects that cover half of the total volume, that is $F(N_{50}) = 50$.

Once the Long Tail is modelled using $F(x)$, we can divide the curve (Figure 5.4) in three parts: head, mid, and the tail. The boundary between the head and the mid part of the curve is defined by:

$$X_{head \to mid} = N_{50}^{2/3} \tag{5.3}$$

Likewise, the boundary between the mid part and the tail is:

$$X_{mid \to tail} = N_{50}^{4/3} \simeq X_{head \to mid}^{2} \tag{5.4}$$

Figure 5.5 depicts the cumulative distribution of the Long Tail from Figure 5.4.

Now, we randomly select 50 artists in each portion of the Long Tail (head, mid and tail), in order to answer these questions: (1) how many artist tags can we classify?, and (2) how many facets do they correspond to?

In order for the Baseline system and our system to be comparable, we use the same set of 8 facets corresponding to a strict match between inferred and expert-defined facets (see Table 5.6).

Table 5.10: Classify Artist tags (the threshold parameter of our system is set to 4).

Long-tail Portion	**Baseline system assigns at least one facet**	**Our system assigns at least one facet**
Head	29.4%	**41.1%**
Mid	25.3%	**33.4%**
Tail	16.9%	**17.1%**

Table 5.11: Average number of assigned facets per artist.

Long-tail Portion	**Baseline system**	**Our system**
Head	1.74	**2.26**
Mid	1.58	**2.16**
Tail	0.84	**1.24**

In order to assign facets to artist tags, we proceed in two steps: (1) for each artist tag, we compute the top facets using Algorithm 5.2, and (2) we merge the tags' top facets using the following equation:

$$\Gamma_{af} = W_{at} \cdot R_{tf} \tag{5.5}$$

where W_{at} represents the weight of tag t for artist a[11], and R_{tf} represents the relevance of facet f for the tag t (R_{tf} is computed using Equation 5.1). For the Baseline system, since it is biased by definition towards the Golden Standard, we did not consider different relevances for facets. In other words, $R_{tf} = 1 \ \forall \ (t, f) \in Baseline$.

Table 5.10 and 5.11 present the results. We can observe that, thanks to attributing a higher number of tags per facet, our method assigns on average more facets to artists than the Baseline system (independently of the popularity of that artist). Yet, the more unknown the artist is the less facets and tags we can assign to her.

Table 5.12: Assignment of facets for artist Elton John's tags.

Tags	**Baseline system**		**Our System**	
	Matched	Facets	Matched	Facets
pop	X	Genre	X	Music_genres, Music_geography
classic rock			X	Music_genres, Record_labels
singer-songwriter			X	Occupations_in_music. Music_genres, Music_performance, Record_labels
rock	X	Genre	X	Music_genres, Music_geography
piano	X	Instrument	X	Musical_instruments, Aspects_of_music
british	X	Locale, country, nationality	X	Music_geography
80s	X	Time period, era / epoch		
70s	X	Time period, era / epoch		
elton john			X	Musicians, Music_people
male vocalists				

An Example

As an illustrative example, we consider the Elton John example, shown in Figure 5.1. Table 5.12 shows the process of assigning facets to artist tags described

[11] this weight is given by the last.fm dataset, and it is computed according to how many times tag t has been applied to artist a. The exact formula, though, is a trade secret of last.fm.

in the previous section. In this example, we fix the threshold parameter of our system to a value of 4.

Both systems are able to classify correctly the tags “pop”, “rock”, “piano” and “british”. The Baseline system was not able to classify “classic rock” nor “singer-songwriter”. This is an illustration of the fact that WordNet does not cover a broad description of the Music domain, a similar problem was reported in (Cano et al., 2004b) about the Sound Effects domain. On the other hand, our system could classify these tags. The tag “singer-songwriter” is classified as `Occupations_in_music`, `Music_genres` and `Music_performance`, which are correct. However, it also classified it as `Record_labels`. A broader analysis of our system's resulting network (as described in Section 5.3) permits us to explore the path between this tag and the facet `Record_labels` (given the threshold limit of 4), namely:

$$Record\_labels \rightarrow Music\_industry \rightarrow Occupations\_in\_music \rightarrow Singer{-}songwriter$$

Similarly, our system classifies the tag “Classic rock” as `Record_labels`. The path there is the following:

$$Record\_labels \rightarrow Music\_industry \rightarrow Radio \rightarrow Radio\_formats \rightarrow Classic\_rock$$

After analyzing the Wikipedia categorization, it appears that there are both *broaderOf* and *subjectOf* links, in both ways, between `Music_industry` and `Record_labels`. While considering that `Music_industry` is a *broaderOf* of `Record_labels`, the opposite does not seem to make sense. This “bug” in the structure of Wikipedia is the reason why an erroneous facet was attributed to these tags. This situation shows us that our system is somehow fragile with respect to noise and inconsistencies in the Wikipedia categorization.

Unlike the Baseline system, our method was not able to classify the tags “70s” and “80s”. This is due to the design characteristics of this system (as in the case of `Moods` described in Section 5.4.3): our method, starting from the seed “Music”, was unable to reach these specific time periods, known as decades, although the facet `Decades` does exist in fact in Wikipedia.

We propose in the next section lines of future work, inspired by (Suchanek et al., 2008), to remedy both this problem and the problem of fragility to noise in the knowledge base.

5.5. Summary and future work

When comparing folksonomies to expert-based taxonomies, the former usually have the advantage to be more complete, and more up-to-date. However, the former have the drawback of lacking structured categories, i.e. terms are not explicitly related to categories.

In this chapter, we focused on the particular domain of knowledge related to Music, and proposed a system addressing the complementary research problems of (1) uncovering the semantic facets of the popular music folksonomy Last.fm, and (2) automatically categorizing music tags according to this set of facets. The Wikipedia repository of knowledge was used as backbone for semi-structured semantic categories.

Our system was able to infer 333 semantic facets of music. By comparing these facets to those of taxonomies defined by experts, we showed that our system is able to cover a significant portion of expert descriptions of music. There were however some expert facets of music that our system could not infer, notably Mood. The main reason is that Mood (or Emotion) is a very generic concept, and can be applied in different contexts. It is then not directly related to music. If our system is exploring explicitely music-related topics, it may not reach this concept. Importantly, we also argued that among the inferred facets, many of those facets that cannot be matched to expert facets are in fact meaningful and do represent valid ways to describe music, at least within the particular realm of the Last.fm folksonomy.

We also showed the relevance of our system in the task of tag categorization on a subset of Gold Standard facets (namely Genre, Instruments, and Locations and Languages), and in the task of automatically categorizing tags of music artists.

There are many avenues for future work. First and foremost regarding evaluations. We intend to proceed to a manual evaluation of the facets inferred by our system that do not correspond to facets commonly found in expert-based taxonomies. We will proceed to this evaluation via a questionnaire-based methodology targeting different participant profiles: experts of the music business, musicians, experts in the scientific research on social music tagging, and lay music lovers.

Further work should also be dedicated to evaluating the worth of the obtained facets and the whole system in a number of tasks, such as music recommendation and browsing, or tag expansion. In particular, we also plan to further study the distributions of music facets with respect to artist popularity and evaluate in what respect our system could be useful for tackling the problem of cold-start in music recommendation (Celma, 2010).

We also plan to modify our system so that it includes the advantages of the Baseline system, without its drawbacks. Namely, instead of starting from the single category “Music”, our system could be based on hierarchies of facets that are known to be relevant to music (through the inclusion of expert-based ontologies), and expand/refine this initial knowledge via specializing the Wikipedia general knowledge base with the same methods as described in this chapter. This may permit to combine in the same system both expert knowledge on specific parts of the universe of music (e.g., Mood taxonomies) with the grounded and updated underlying knowledge of music lovers (Suchanek et al., 2008).

Last, but not least, we plan to study different measures for evaluating the relevance of a music facet to a given tag. This would improve the precision of our system in tasks such as tag classification.

The whole system, its data and source code are available on the website http://www.dtic.upf.edu/~msordo/wikifacets in order to stimulate its use by fellow researchers.

CHAPTER 6

Conclusions and Future Work

At first, science has a bitter taste, but at the end it is sweeter than honey.
—The brethren of purity — Ikhwan Al–Safa (10th Century)

6.1. Introduction

When this thesis started, there was almost no published work related with music autotagging, although a number of research works using contextual information for music mining and classification (Celma et al., 2006; Ellis et al., 2002; Geleijnse et al., 2007; Knees et al., 2006, 2007a; Schedl et al., 2005a,b, 2006; Whitman & Ellis, 2004; Whitman & Lawrence, 2002; Whitman & Rifkin, 2002) were available. The first autotagging approaches (Eck et al., 2007; Turnbull et al., 2006) were built upon previous specific Music Information Retrieval tasks, such as genre, mood or artist classification. Since then, several contributions have been made to the field, including data gathering, machine learning algorithms, and methological form (Marques et al., 2011). Nowadays, tags are one of the key subjects in the Music Information Retrieval field (Downie, 2003).

In Chapter 2 we presented the formalization and framework of music automatic tagging, referencing related work. In Chapter 3 we introduced our proposed automatic music tagging approach. The algorithm predicts tags based on acoustic similarity, using a memory-based classifier and a pre-defined labeled training dataset. The audio features used in this thesis cover a wide range of musical concepts, including timbre (e.g., MFCC), tonal (e.g., pitch class profiles), temporal (e.g., bpm, onset, peak weights) and a set of high level features, such as moods, genres, etc. In order to assess the strength of the proposed autotagging algorithm, we carried out a thorough evaluation in several experiments, using six datasets and different methods for music annotation. Experimental results revealed that a simple model, combined with an audio feature representation that covers a variety of music concepts can perform as well as, or better than many state of the art approaches. Interestingly, many of the aforementioned state of the art approaches use much more complex, time and resource consuming algorithms, though they rely only on timbre information. These results are additional evidence that special care must be taken in selecting and capturing more complete and descriptive audio–related information, in order to build successful models for automatic tagging of music (Herrera-Boyer et al., 2006).

The last few years have also shown a significant increase in social tagging. Since 2004, the explosion of Web 2.0 (e.g., tagging, blogging, user–generated content, etc.) questioned the usefulness of controlled vocabularies (Shirky, 2005). Internet sites with a strong social component, like *Last.fm*, allow users to tag music according to their own criteria. This scenario made the world of taxonomies even more complex. Nowadays, users can organize their music collection using social tags like *late night*, *while driving*, etc. The combination of tags from thousands of users lead to the emergence of a large body of domain-specific knowledge, often referred to as *Folksonomy*. Folksonomies exploit user–generated classification through a bottom–up approach (Scaringella et al., 2006). On the one hand, this non-hierarchical approach allows users to organize their music with a better confidence. On the other hand, it creates difficulties for the design and maintenance of expert–based taxonomies, as new terms may emerge from time to time. Thus, in this scenario, up to date expert–based taxonomies become more and more difficult. Moreover, the simplicity and user-friendliness of community–based tagging imposes a toll: there is usually no way to *explicitly* relate tags with the corresponding music facets. When browsing the tag description of a particular artist, *Last.fm* users browse a —albeit very rich— *flat* list of terms.

We tackled these last issues in the chapters 4 and 5. In Chapter 4 we studied whether the controlled vocabulary defined by a group of experts correspond with the tag annotations of a large community, the so–called wisdom of the crowds. We ran experiments in two basic musical concepts: music genre and mood. Regarding music genre, we found that some genres are clearly defined both from the experts and the wisdom of crowds, reaching a high agreement between these two views, while other genres are difficult to get a common consensus of its meaning. As for moods, we demonstrated that the basic emotions *happy*, *sad*, *angry* and *tender* are very relevant to the social community. With respect to expert–defined mood clustering representations, we found that the arousal and valence dimensions based on Rusell's model of emotion (Russell, 1980) can also be captured.

In Chapter 5, we proposed a system addressing the complementary research problems of: uncovering the semantic facets of the popular music folksonomy Last.fm, and automatically categorizing music tags according to this set of facets. The Wikipedia repository of knowledge was used as a backbone for semi-structured semantic categories. Our system was able to infer 333 semantic facets of music. By comparing these facets to those of taxonomies defined by experts, we showed that our system is able to cover a significant portion of expert descriptions of music. There were however some expert facets of music that our system could not infer, notably Mood. The main reason is that Mood (or Emotion) is a very generic concept, and can be applied in different contexts. It is then not directly related to music. If our system is exploring explicitely music-related topics, it may not reach this concept. Importantly, we also argued that among the inferred facets, many of those facets that cannot be matched to expert facets are in fact meaningful and do represent valid ways to describe music, at least within the particular realm of the Last.fm folksonomy. We also showed the relevance of our system in the task of tag categorization on a subset of Gold Standard facets (namely Genre, Instruments, and Locations and Languages), and in the task of automatically categorizing tags of music artists.

6.2. Summary of contributions

The main contributions of this thesis are:

1. An automatic music tagging algorithm that uses acoustic similarity and nearest neighbor classification to propagate tags among songs. The algorithm has the following advantages:

    a) It avoids the design and training of each possible tag, specially for datasets based on folksonomies, where there are thousands of tags.

    b) From the industry perspective, it shows to be scalable in both memory and CPU time consumption, for datasets in the order of tens of thousands of tags and hundreds of thousands of music excerpts.

2. An extensive evaluation of the autotagging algorithm using multiple datasets, for both music and sound effects. It compares the experimental results with several approaches that are representative of the state of the art of music autotagging. Additionally, it discusses and emphasizes the importance of different evaluation measures, local or global, for the task of music autotagging.

3. An in-depth study of the nature of music folksonomies, focusing on the social music website *Last.fm*. The aim of such study is to assess whether the tag annotations made by a large community concord with a controlled, structured vocabulary of experts in their field, by reconstructing taxonomies from the inherent correlation amongst the semantic terms (tags). This study focuses on two main aspects of music: musical genres and moods.

4. A generalization of the previous contribution to a wide range of semantic concepts. This thesis presents a novel way to uncover a set of semantic

facets implicit in the tags of the *Last.fm* music folksonomy, and classify tags with respect to these facets, using the semi-structured repository of general knowledge *Wikipedia* as a backbone for tag categorization.

6.3. Limitations and future work

The problem of annotating music with semantic words is far from being solved. There are many open issues and avenues for future work.

Consistent datasets and labeled data. As Bertin-Mahieux et al. (2010) state, current research in music autotagging is moving towards the use of large datasets, principally retrieved from web documents or social data. The rationale is that large datasets can help to overcome the inconsistencies inherent in the tagging vocabulary. In this thesis, however, it is presented experimentally that the problem of annotating music from a large dataset of social tags, is far from being solved. Datasets usually suffer from data scarcity or tag correlation (similarity, polysemy). Moreover, social tags can evolve over time, and new or more complex concepts can appear. One of the contributions of this thesis consisted in building an automatic model to uncover the semantic facets inherent to social tags. The semantic facets are anchored upon the structure of Wikipedia, a dynamic and evolving encyclopedia repository of universal knowledge. Interestingly, this aspect of evolution can help overcome the problem of new, emerging concepts in tagging. A clear plan for future work is to include these models in the proposed autotagging algorithm.

Features and learning algorithms. Selecting the best set of features that discriminate different objects in a classification task is crucial. Indeed, Herrera-Boyer et al. (2006) state that "*choosing good features is more crucial than the choice of the classification algorithm, and the classification itself becomes easier if the features chosen are informative enough*". It should be noted, though, that tag classification covers much more concepts than traditional MIR classification tasks, including mood, genre, or artist classification. For instance, genre and mood are only 2 facets among many others. Future work includes selecting features that discriminate each concept separately, or uncovering features that are common for a combination of concepts. The latter approach can be helpful for solving the problem of tag correlation. Further work should be devoted also to the way features are aggregated, beyond the classical bag of frames approach (Aucouturier et al., 2007a; Seyerlehner, 2010; Seyerlehner et al., 2010).

Cross–collection tagging. Traditional research in MIR related Machine Learning tasks consists of using the same Ground Truth dataset for learning models and testing their quality (by performing any validation method). Marques et al. (2011) remark that using a dataset for training and another different dataset for testing negatively influences the evaluation results, and that special care must be taken to understand what are we learning. An additional line of research may include the use of different datasets for training. A high level autotagger can be built by aggregating several autotaggers trained with a different dataset, as it has been recently addressed by Ellis et al. (2011). In addition, each dataset can be defined for a specific music concept (moods, genres, usage, etc.), or alternatively with overlapping concepts which can help to reinforce the classification or misclassification of an audio excerpt. The main drawback of this latter approach is that datasets from different sources tend to share few concepts or use different words for expressing the same concepts. Further work on tag similarity should help overcome these shortcomings.

Evaluation. Several experiments in this thesis have revealed that there is no unique way to assess the quality of autotagging algorithms. Some algorithms perform well in per–song evaluations, however they fail to predict all the tags, especially those that were less frequently used, hence resulting in worse per–tag evaluations. Other algorithms are, a priori, more robust at learning models for all the tags and predict most of the tags, which results in higher per–tag but lower per–song performance. These evaluations are, however, very generic, and do not uncover the particularities of each autotagging system. For this purpose, individual evaluations for each tag were also considered. Furthermore, the characteristics of each autotagging approach can be more or less valuable depending on the application of the "autotags" (e.g., search, similarity, recommendation). Future work includes a subjetive evaluation of the algorithm predictions, although it should be considered the current impracticability for such evaluation, especially for large datasets. Another alternative that can be considered is using the proposed annotations in a higher level task such as music recommendation (Eck et al., 2008; Zhao et al., 2010). Among many other aspects, Urbano (2011) states that evaluation should go beyond the comparison of evaluation results. The author exposes a wide list of recommendations for improving MIR related algorithms, including standardized collections, baselines or evaluation models.

For semantic facets and concepts, we intend to proceed to a manual evaluation of the facets inferred by our system that do not correspond to facets commonly found in expert-based taxonomies. We will proceed to this evaluation via a questionnaire-based methodology targeting different participant profiles: experts of the music business, musicians, experts in the scientific research on social music tagging, and lay music lovers.

Refining the music semantic facets. Regarding the presented algorithm for uncovering music facets, instead of starting from the single category "Music", our system could be based on hierarchies of facets that are known to be relevant to music (through the inclusion of expert–based ontologies). This may permit combining in the same system both expert knowledge on specific parts of the universe of music (e.g., Mood taxonomies) with the grounded and updated underlying knowledge of music lovers (Suchanek et al., 2008). Finally, we plan to study different measures for evaluating the relevance of a music facet to a given tag. This would improve the precision of our system in tasks such as tag classification.

Hybrid approaches. Last but not least, we strongly believe that neither a pure content–based approach nor a system that relies only on contextual information (e.g., social tags) can solve the problem of tagging music separately. Each one of these approaches has advantages and shortcomings. A multi–faceted approach using expert based classifications, dynamic associations derived from the community driven annotations, and content–based analysis would improve audio tag classification. Some work has already been done for combining such sources of information (Barrington et al., 2009; Knees et al., 2009), with promising results. Future work can be carried out regarding how to effectively combine these diverse sources of information.

I hope you enjoyed reading this thesis.

Mohamed Sordo, Barcelona, December 20, 2011.

Bibliography

Aggarwal, C. C. (2005). On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pp. 901–909. VLDB Endowment.

Agostini, G., Longari, M., & Pollastri, E. (2003). Musical instrument timbres classification with spectral features. *EURASIP Journal on Applied Signal Processing*, *2003*, 5–14.

Alpaydin, E. (2004). *Introduction to machine learning.* The MIT Press.

Ames, M. & Naaman, M. (2007). Why we tag: motivations for annotation in mobile and online media. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 971–980.

Aucouturier, J. & Defreville, B. (2009). Judging the similarity of soundscapes does not require categorization: Evidence from spliced stimuli. *The Journal of the Acoustical Society of America*, *125*, 2155.

Aucouturier, J., Defreville, B., & Pachet, F. (2007a). The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, *122*, 881.

Aucouturier, J., Pachet, F., Roy, P., & Beurivé, A. (2007b). Signal+context= better classification. In *Proceedings of the International Conference for Music Information Retrieval.* Vienna, Austria.

Aucouturier, J.-J. (2009). Sounds like teen spirit: Computational insights into the grounding of everyday musical terms. In J. Minett & W. Wang (Eds.) *Language, Evolution and the Brain, Frontiers in Linguistics Series.* Taipei: Academia Sinica Press.

Aucouturier, J.-J. & Pachet, F. (2004). Improving timbre similarity: how high's the sky. In *Journal of Negative Results in Speech and Audio Science.*

Baeza-Yates, R., Ribeiro-Neto, B. et al. (1999). *Modern information retrieval*, vol. 82. Addison-Wesley New York.

Barrington, L., Turnbull, D., Yazdani, M., & Lanckriet, G. (2009). Combining audio content and social context for semantic music discovery. *Proceedings of the 32nd ACM SIGIR conference on Research and development in information retrieval.*

Beckett, D. (2006). Semantics through the tag. In *Proceedings of the XTech Conference*. Amsterdam, The Netherlands.

Bellegarda, J. (Sept. 2005). Latent semantic mapping. *Signal Processing Magazine, IEEE*, *22*(5), 70–80.

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

Bertin-Mahieux, T., Eck, D., & Mandel, M. (2010). Automatic tagging of audio: The state-of-the-art. In W. Wang (Ed.) *Machine Audition: Principles, Algorithms and Systems*. IGI Global. In press.

Bertin-Mahieux, T., Ellis, D., Whitman, B., & Lamere, P. (2011). The million song dataset. In *Proceedings of the International Conference on Music Information Retrieval*. Miami, FL, USA.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In *International Conference on Database Theory*, pp. 217–235.

Bischoff, K., Firan, C., C., K., Nejdl, W., & Paiu, R. (2009). Automatically identifying tag types. In *Proceedings of the fifth International Conference on Advanced Data Mining and Applications*, pp. 31–42. Beijing, China.

Bogdanov, D., Serrà, J., Wack, N., Herrera, P., & Serra, X. (2011). Unifying low-level and high-level music similarity measures. *IEEE Transactions on Multimedia*, *13*, 687–701.

Box, G. & Cox, D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211–252.

Buckma, J. (2004). Magnatune, an open music experiment. *Linux Journal*, (118), 42–44.

Burred, J. J. & Peeters, G. (2009). An adaptive system for music classification and tagging. In S. Baumann, J. J. Burred, A. Nürnberger, & S. Stober (Eds.) *Proceedings of the 3rd Workshop on Learning the Semantics of Audio Signals*, pp. 3–16. Graz, Austria.

Cano, P. (2007). *Content-Based Audio Search from Fingerprinting to Semantic Audio Retrieval*. Ph.D. thesis, Universitat Pompeu Fabra.

Cano, P. & Koppenberger, M. (2004). Automatic sound annotation. In *Proceedings of 14th IEEE workshop on Machine Learning for Signal Processing*. São Luís, Brazil.

Cano, P., Koppenberger, M., Ferradans, S., Martinez, A., Gouyon, F., Sandvold, V., Tarasov, V., & Wack, N. (2004a). Mtg-db: A repository for music audio processing. In *Proceedings of 4th International Conference on Web Delivering of Music*. Barcelona, Spain.

Cano, P., Koppenberger, M., Herrera, P., Celma, O., & Tarasov, V. (2004b). Sound effect taxonomy management in production environments. In *Proceedings of 25th International AES Conference*. London, UK.

Cano, P., Koppenberger, M., Le Groux, S., Ricard, J., Wack, N., & Herrera, P. (2005). Nearest-neighbor automatic sound classification with a wordnet taxonomy. *Journal of Intelligent Information Systems*, *24*(2), 99–111.

Carneiro, G. & Vasconcelos, N. (2005). Formulating Semantic Image Annotation as a Supervised Learning Problem. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, *2*.

Casey, M., Rhodes, C., & Slaney, M. (2008). Analysis of minimum distances in high-dimensional musical spaces. *Audio, Speech, and Language Processing, IEEE Transactions on*, *16*(5), 1015–1028.

Celma, O. (2006). *Music Recommendation: a multi-faceted approach*. Master's thesis, Universitat Pompeu Fabra.

Celma, O. (2010). *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer.

Celma, O., Cano, P., & Herrera, P. (2006). Search sounds: An audio crawler focused on weblogs. In *Proceedings of 7th Intl. Conference on Music Information Retrieval*. Victoria, Canada.

Celma, Ò. & Lamere, P. (2007). Music recommendation tutorial. In *International Conference on Music Information Retrieval*. Vienna, Austria.

Chang, C., Lin, C. et al. (2001). Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chen, S. S., Setauket, E., & Gopinath, R. A. (2001). Gaussianization.

Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2007). Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Reviews*, *51*, 661–703.

Collins, M., Dasgupta, S., & Schapire, R. (2001). A generalization of principal component analysis to the exponential family. In *Neural Information Processing Systems*. Vancouver, British Columbia, Canada.

Cook, P. (2001). *Music, cognition, and computerized sound: an introduction to psychoacoustics*. The MIT Press.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, *13*(1), 21–27.

Coviello, E., Chan, A., & Lanckriet, G. (2010). Time series models for semantic music annotation. *Audio, Speech, and Language Processing, IEEE Transactions on*, *19*, 1343–1359.

Coviello, E., Miotto, R., & Lanckriet, G. (2011). Combining content-based auto-taggers with decision-fusion. In *Proceedings of the International Society for Music Information Retrieval Conference*. Miami, FL, USA.

Csikszentmihalyi, M. (1997). *Finding flow: The psychology of engagement with everyday life*. Basic Books (AZ).

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Dong, J., Krzyzak, A., & Suen, C. (2005). Fast svm training algorithm with decomposition on very large data sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*(4), 603–618.

Downie, J. (2003). Music information retrieval. *Annual review of information science and technology*, *37*(1), 295–340.

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

Downie, J., Byrd, D., & Crawford, T. (2009). Ten years of ismir: Reflections on challenges and opportunities. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pp. 13–18.

Duan, Z., Lu, L., & Zhang, C. (2008). Collective annotation of music from multiple semantic categories. In *International Conference on Music Information Retrieval*. Philadelphia, PA, USA.

Eck, D., Bertin-Mahieux, T., & Lamere, P. (2007). Autotagging music using supervised machine learning. In *Proceedings of the 8th International Conference on Music Information Retrieval*.

Eck, D., Lamere, P., Bertin-Mahieux, T., & Green, S. (2008). Automatic generation of social tags for music recommendation. In *Advances in Neural Information Processing Systems 20*, pp. 385–392. Cambridge, MA: MIT Press.

Ellis, D., Berenzweig, A., & Whitman, B. (2003). The uspop2002 pop music data set. http://www.ee.columbia.edu/~dpwe/research/musicsim/uspop2002.html.

Ellis, D., Whitman, B., A.Berenzweig, & S.Lawrence (2002). The quest for ground truth in musical artist similarity. In *Proceedings of 3rd International Symposium on Music Information Retrieval*, pp. 170–177. Paris.

Ellis, K., Coviello, E., & Lanckriet, G. (2011). Semantic annotation and retrieval of music using a bag of systems representation. In *Proceedings of the International Society for Music Information Retrieval*. Miami, FL, USA.

Eriksson, L., Johansson, E., Muller, M., & Wold, S. (2000). On the selection of the training set in environmental qsar analysis when compounds are clustered. *Journal of Chemometrics*, *14*, 599–616.

Fabbri, F. (1982). A theory of musical genres: Two applications. *Popular Music Perspectives*, *1*, 52–81.

Feng, Y., Zhuang, Y., & Pan, Y. (2003). Music information retrieval by detecting mood via computational media aesthetics. In *The 2003 IEEE/WIC International Conference on Web Intelligence*, pp. 235–241. Halifax, Canada.

Flexer, A. (2006). Statistical evaluation of music information retrieval experiments. *Journal of New Music Research*, *35*(2), 113–120.

Flexer, A. (2007). A closer look on artist filters for musical genre classification. *World*, *19*(122), 16–7.

Freesound.org (2011). Analysis descriptors documentation. http://www.freesound.org/docs/api/analysis_docs.html. Last accessed Oct. 2011.

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Frijda, N. (1986). *The emotions: Studies in emotion and social interaction*, vol. 44. Cambridge University Press.

Fujishima, T. (1999). Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the International Computer Music Conference, 1999*, pp. 464–467. Beijing, China.

Gaver, W. W. (1993). What in the world do we hear? an ecological approach to auditory event perception. *Ecological Psychology*, *5*(1), 1–29.

Geleijnse, G., Schedl, M., & Knees, P. (2007). The quest for ground truth in musical artist tagging in the social web era. In *Proceedings of the Eighth International Conference on Music Information Retrieval*, pp. 525 – 530. Vienna, Austria.

Göker, A. & Myrhaug, H. (2002). User context and personalisation. In *European Conference on Case Based Reasoning.*

Gomez, E. (2006). *Tonal Description of Music Audio Signals.* Ph.D. thesis, Universitat Pompeu Fabra.

Goutte, C. (1997). Note on free lunches and cross-validation. *Neural Computation*, *9*, 1211–1215.

Gouyon, F. (2005). *A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing.* Ph.D. thesis, Universitat Pompeu Fabra.

Gouyon, F., Herrera, P., Gómez, E., Cano, P., Bonada, J., Loscos, A., Amatriain, X., & Serra, X. (2008). *Content Processing of Music Audio Signals*, chap. 3, pp. 83–160. Logos Verlag Berlin GmbH.

Guaus, E. (2009). *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers.* Ph.D. thesis, Universitat Pompeu Fabra.

Hall, M. A. (1998). *Correlation-based Feature Subset Selection for Machine Learning.* Hamilton, New Zeland.

Hamel, P., Lemieux, S., Bengio, Y., & Eck, D. (2011). Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proceedings of the International Society for Music Information Retrieval Conference.* Miami, FL, USA.

Han, E. & Karypis, G. (2000). Centroid-based document classification: Analysis and experimental results. *Principles of Data Mining and Knowledge Discovery*, pp. 116–123.

Harman, D. (1993). Overview of trec-1. In *Proceedings of the workshop on Human Language Technology*, pp. 61–65. Association for Computational Linguistics.

Herrera, P., Celma, O., Massaguer, J., Cano, P., Gómez, E., Gouyon, F., Koppenberger, M., Garcia, D., G. Mahedero, J., & Wack, N. (2005). Mucosa: a music content semantic annotator. In *Proceedings of 6th International Conference on Music Information Retrieval.* London, UK.

Herrera-Boyer, P., Klapuri, A., & Davy, M. (2006). Automatic classification of pitched musical instrument sounds. *Signal processing methods for music transcription*, pp. 163–200.

Herrera-Boyer, P., Peeters, G., & Dubnov, S. (2003). Automatic classification of musical instrument sounds. *Journal of New Music Research*, *32*(1), 3–21.

Hevner, K. (1936). Experimental studies of the elements of expression im music. *The American Journal of Psychology*, *48*(2), 246–268.

Hinkelmann, K., Kempthorne, O., & Wiley, J. (2005). *Design and analysis of experiments*, vol. 1. Wiley Online Library.

Hoffman, M., Blei, D., & Cook, P. (2009). Easy as cba: A simple probabilistic model for tagging music. In *Proceedings of the International Conference on Music Information Retrieval*. Kobe, Japan.

Hu, X., Downie, J., & Ehmann, A. (2009). Lyric text mining in music mood classification. *American music*, *183*(5,049), 2–209.

Hu, X., Downie, J., Laurier, C., Bay, M., & Ehmann, A. (2008). The 2007 mirex audio mood classification task: Lessons learned. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pp. 462–467. Citeseer.

IMIRSEL (2011a). Mirex 2011 audio tag classification. `http://www.music-ir.org/mirex/wiki/2011:Audio_Tag_Classification`. Last accessed Oct. 2011.

IMIRSEL (2011b). Mirex 2011 results. `http://www.music-ir.org/mirex/wiki/2011:MIREX2011_Results`. Last accessed Oct. 2011.

Jaynes, E. & Bretthorst, G. (2003). *Probability theory: the logic of science*. Cambridge Univ Press.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edn.

Jones, M., Downie, J., & Ehmann, A. (2007). Human similarity judgments: Implications for the design of formal evaluations. In *Proceedings of ISMIR 2007 International Society of Music Information Retrieval*.

Juslin, P. N. & Sloboda, J. A. (2001). *Music and Emotion: Theory and Research*. Oxford University Press.

Kilkki, K. (2007). A practical model for analyzing long tails. *First Monday*, *12*(5).

Kim, H., Howland, P., & Park, H. (2006). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research*, *6*(1), 37.

Kim, Y., Schmidt, E., & Emelle, L. (2008). Moodswings: A collaborative game for music mood label collection. In *Proceedings of the International Conference on Music Information Retrieval*, pp. 231–236. Philadelphia, PA, USA.

Knees, P. (2007). Search & Select - Intuitively Retrieving Music from Large Collections. In *Proceedings of the 8th International Conference on Music Information Retrieval*. Vienna, Austria.

Knees, P. (2010). *Text-Based Description of Music for Indexing, Retrieval, and Browsin*. Ph.D. thesis, Johannes Kepler University.

Knees, P., Pohle, T., Schedl, M., Schnitzer, D., & Seyerlehner, K. (2008). A document-centered approach to a natural language music search engine. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, pp. 627–631. Springer-Verlag.

Knees, P., Pohle, T., Schedl, M., Schnitzer, D., Seyerlehner, K., & Widmer, G. (2009). Augmenting text-based music retrieval with audio similarity. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*. Kobe, Japan.

Knees, P., Pohle, T., Schedl, M., & Widmer, G. (2006). Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval*. Santa Barbara, California, USA.

Knees, P., Pohle, T., Schedl, M., & Widmer, G. (2007a). A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Amsterdam, the Netherlands.

Knees, P., Pohle, T., Schedl, M., & Widmer, G. (2007b). A music search engine built upon audio-based and web-based similarity measures. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 447–454. ACM.

Knees, P. & Widmer, G. (2007). Searching for Music Using Natural Language Queries and Relevance Feedback. In *Proceedings of the 5th International Workshop on Adaptive Multimedia Retrieval*. Paris, France.

Korn, F., Pagel, B.-U., & Faloutsos, C. (2001). On the “dimensionality curse” and the “self-similarity blessing”. *Knowledge and Data Engineering, IEEE Transactions on*, *13*(1), 96 –111.

Lamere, P. (2008). Social tagging and Music Information Retrieval. *Journal of New Music Research*, *37*(2), 101–114.

Lamere, P. & Pampalk, E. (2008). Social tags and music information retrieval. In *International Conference on Music Information Retrieval*. Philadelphia, PA, USA.

Laurier, C. (2011). *Automatic Classification of Musical Mood by Content-Based Analysis*. Ph.D. thesis, Universitat Pompeu Fabra.

Laurier, C. & Herrera, P. (2009). Automatic detection of emotion in music: Interaction with emotionally sensitive machines. *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*, pp. 9–32.

Laurier, C., Meyers, O., Serra, J., Blech, M., & Herrera, P. (2009a). Music mood annotator design and integration. In *Content-Based Multimedia Indexing, 2009. Seventh International Workshop on*, pp. 156–161. Chania, Crete, Grece: IEEE.

Laurier, C., Sordo, M., Serrà, J., & Herrera, P. (2009b). Music mood representations from social tags. In *International Conference on Music Information Retrieval*. Kobe, Japan.

Law, E., von Ahn, L., & Dannenberg, R. (2007). Tagatune: a game for music and sound annotation. In *Proceedings of the International Conference on Music Information Retrieval*. Vienna, Austria.

Lehmann, E. & Romano, J. (2005). *Testing statistical hypotheses*. Springer Verlag.

Levy, M. & Sandler, M. (2007). A semantic space for music derived from social tags. In *Proceedings of the Eighth International Conference on Music Information Retrieval*. Vienna, Austria.

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

Levy, M. & Sandler, M. (2009). Music information retrieval using social tags and audio. *Multimedia, IEEE Transactions on*, *11*(3), 383–395.

Li, T. & Ogihara, M. (2003). Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 239–240. Baltimore, Maryland, USA.

Liu, D., Lu, L., & Zhang, H. (2003). Automatic mood detection from acoustic music data. In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 81–87. Baltimore, Maryland, USA.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *Proceedings of 1st International Conference on Music Information Retrieval.* Plymouth, MA.

Mahalanobis, P. (1936). On the generalized distance in statistics. In *Proceedings of the National Institute of Science, Calcutta*, vol. 12, p. 49.

Mandel, M. & Ellis, D. (2005). Song-level features and support vector machines for music classification. In *Proceedings of the International Conference on Music Information Retrieval.* London, UK.

Mandel, M. & Ellis, D. (2007). A web-based game for collecting music metadata. In *Proceedings of the International Conference on Music Information Retrieval.* Vienna, Austria.

Mandel, M. & Ellis, D. (2008). Multiple-instance learning for music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval*, pp. 577–582. Philadelphia, PA, USA.

Mandel, M. I., Pascanu, R., Eck, D., Bengio, Y., Aiello, L. M., Schifanella, R., & Menczer, F. (2011). Contextual tag inference. *ACM Trans. Multimedia Comput. Commun. Appl.*, *7S*, 32:1–32:18.

Mani, I. (2002). Automatically inducing ontologies from corpora. In *Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology.* Geneva, Switzerland.

Marques, G., Domingues, M., Langlois, T., & Gouyon, F. (2011). Three current issues in music autotagging. In *International Society for Music Information Retrieval Conference.* Miami, FL, USA.

Martens, H. A. & Dardenne, P. (1998). Validation and verification of regression in small data sets. *Chemometrics and Intelligent Laboratory Systems*, *44*, 99–121.

McKay, C. & Fujinaga, I. (2004). Automatic genre classification using large high-level musical feature sets. In *Proceedings of the International Conference on Music Information Retrieval*, vol. 525, p. 30. Barcelona, Spain.

Mckay, C. & Fujinaga, I. (2006). Musical genre classification: Is it worth pursuing and how can it be improved? In *Proceedings of the Seventh International Conference on Music Information Retrieval.* Victoria, Canada.

Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, *Lecture Notes in Computer Science*, vol. 3729, pp. 522–536. International Semantic Web Conference 2005, Springer.

Miller, G. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, *63*(2), 81.

Miotto, R., Barrington, L., & Lanckriet, G. (2010). Improving auto-tagging by modeling semantic co-occurrences. In *Proceedings of the International Society for Music Information Retrieval Conference*. Utrecht, The Nederlands.

Ness, S., Theocharis, A., Tzanetakis, G., & Martins, L. (2009). Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proceedings of the seventeen ACM international conference on Multimedia*, pp. 705–708. ACM.

Orio, N. (2006). Music retrieval: a tutorial and review. *Foundations and Trends in Information Retrieval*, *1*, 1–96.

Overell, S., Sigurbjörnsson, B., & van Zwol, R. (2009). Classifying tags using open content resources. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 64–73. Barcelona, Spain: ACM.

Pachet, F. (2005). Knowledge Management and Musical Metadata. *Encyclopedia of Knowledge Management.*

Pachet, F. & Cazaly, D. (2000). A taxonomy of musical genres. In *Content-Based Multimedia Information Access Conference (RIAO)*. Paris, France.

Pachet, F. & Roy, P. (2009). Improving multi-label analysis of music titles: A large scale validation of the correction approach. *IEEE Transactions on Audio Speech and Language Processing*, *2*(17), 335–343.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, J., Taylor, S., & Thomas, E. (2009). Reducing ambiguity in tagging systems with folksonomy search expansion. *The Semantic Web: Research and Applications*, pp. 669–683.

Panagakis, Y., Kotropoulos, C., & Arce, G. (2010). Sparse multilabel linear embedding within nonnegative tensor factorization applied to music tagging. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 393–398. Utrecht, the Netherlands.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. In *Proocedings of the ACM Conference on Principles of Database Systems (PODS)*, pp. 159–168. Seattle.

Park, H., Jeon, M., & Rosen, J. (2003). Lower dimensional representation of text data based on centroids and least squares. *BIT Numerical mathematics*, *43*(2), 427–448.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, *10*(3), 61–74.

Quinlan, R. (1986). Induction of decision trees. *Machine Learning Journal*, *1*, 81–106.

Rabiner, L. & Juang, B. (1993). *Fundamentals of speech recognition*. Prentice hall.

Raimond, Y., Giasson, F., Jacobson, K., Fazekas, G., Gangler, T., & Reinhardt, S. (2010). *Music Ontology Specification*. http://musicontology.com/. Last accessed Oct. 2010.

Ratcliff, J. & Metzener, D. (1988). Pattern matching: The Gestalt approach. *Dr. Dobb's Journal*, p. 46.

Rauber, A., Pampalk, E., & Merkl, D. (2002). Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity. In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 71–80.

Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, *190*, 131–154.

Russell, J. (1980). A circumplex model of affect. *Journal of personality and social psychology*, *39*(6), 1161.

Samet, H. (2006). *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.

Sarmento, L., Gouyon, F., & Oliveira, E. (2009). Music artist tag propagation with wikipedia abstracts. In *Workshop on Information Retrieval over Social Networks, European Conference on Information Retrieval*. Toulouse, France.

Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, *23*(2), 133–141.

Schaeffer, P. (1966). *Trait des objects musicaux.* Paris: Editions du Seuil.

Schedl, M. & Knees, P. (2011). Personalization in multimodal music retrieval. In *Proceedings of the 9th Workshop on Adaptive Multimedia Retrieval.*

Schedl, M., Knees, P., & Widmer, G. (2005a). Improving prototypical artist detection by penalizing exorbitant popularity. In *Proceedings of 3rd International Symposium on Computer Music Modeling and Retrieval*, pp. 196–200. Pisa, Italy.

Schedl, M., Knees, P., & Widmer, G. (2005b). A web-based approach to assessing artist similarity using co-occurrences. In *Proceedings of 4th International Workshop on Content-Based Multimedia Indexing.* Riga, Latvia.

Schedl, M., Knees, P., & Widmer, G. (2006). Investigating Web-Based Approaches to Revealing Prototypical Music Artists in Genre Taxonomies. In *Proceedings of the 1st IEEE International Conference on Digital Information Management (ICDIM'06).* Bangalore, India.

Schedl, M. & Pohle, T. (2010). Enlightening the sun: A user interface to explore music artists via multimedia content. *Multimedia Tools and Applications: Special Issue on Semantic and Digital Media Technologies*, *49*, 101–118.

Schmitz, P. (2006). Inducing ontology from flickr tags. In *Proceedings of the Workshop on Collaborative Tagging at WWW2006.* Edinburgh, Scotland.

Schölkopf, B., Smola, A., Williamson, R., & Bartlett, P. (2000). New support vector algorithms. *Neural computation*, *12*(5), 1207–1245.

Schuth, A., Marx, M., & de Rijke, M. (2007). Extracting the discussion structure in comments on news-articles. In *9th ACM International Workshop on Web Information and Data Management*, pp. 97–104. Lisboa, Portugal.

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

Seyerlehner, K. (2010). *Content-based Music Recommender Systems: Beyond simple Frame-Level Audio Similarity.* Ph.D. thesis, Johannes Kepler University.

Seyerlehner, K., Widmer, G., Schedl, M., & Knees, P. (2010). Automatic music tag classification based on block-level. In *Proceedings of the Sound and Music Computing Conference*. Barcelona, Spain.

Shental, N., Hertz, T., Weinshall, D., & Pavel, M. (2006). Adjustment learning and relevant component analysis. *Computer Vision—ECCV 2002*, pp. 181–185.

Shirky, C. (2005). Ontology is overrated: Categories, Links, and Tags. http://shirky.com/writings/ontology_overrated.html.

Sigurbjörnsson, B. & van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *Proceeding of the 17th international conference on World Wide Web*, pp. 327–336. Beijing, China: ACM.

Slaney, M. (2002). Semantic-audio retrieval. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 4, pp. IV–4108 –IV–4111.

Sordo, M., Celma, O., Blech, M., & Guaus, E. (2008). The quest for musical genres: Do the experts and the wisdom of crowds agree? In *International Conference on Music Information Retrieval*. Philadelphia, PA, USA.

Sordo, M., Gouyon, F., & Sarmento, L. (2010). A method for obtaining semantic facets of music tags. In *1st Workshop On Music Recommendation And Discovery, ACM RecSys, 2010, Barcelona, Spain*. Barcelona.

Sordo, M., Laurier, C., & Celma, O. (2007). Annotating music collections: How content-based similarity helps to propagate labels. In *Proceedings of 8th International Conference on Music Information Retrieval*. Vienna, Austria.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society, B*(38), 44–47.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2008). YAGO: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, *6*(3), 203–217.

Tague-Sutcliffe, J. & Blustein, J. (1995). A statistical analysis of the trec-3 data. *Overview of the Third Text Retrieval Conference (Trec-3)*.

Tingle, D., Kim, Y., & Turnbull, D. (2010). Exploring automatic music annotation with acoustically-objective tags. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pp. 55–62. Philadelphia, PA, USA: ACM.

Torres, D., Turnbull, D., Barrington, L., & Lanckriet, G. (2007). Identifying words that are musically meaningful. In *Proceedings of 8th International Conference on Music Information Retrieval*. Vienna, Austria.

Tsang, I., Kwok, J., & Cheung, P. (2006). Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, *6*(1), 363.

Turnbull, D., Barrington, L., & Lanckriet, G. (2006). Modelling music and words using a multi-class naıve bayes approach. In *Proceedings of the International Conference on Music Information Retrieval*. Victoria, British Columbia, Canada.

Turnbull, D., Barrington, L., & Lanckriet, G. (2008a). Five approaches to collecting tags for music. In *International Conference on Music Information Retrieval*. Philadelphia, PA, USA.

Turnbull, D., Barrington, L., Torres, D., & Lanckriet, G. (2007a). Towards musical query-by-semantic-description using the CAL500 data set. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 439–446.

Turnbull, D., Barrington, L., Torres, D., & Lanckriet, G. (2008b). Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, *2*(16), 467–476.

Turnbull, D., Liu, R., Barrington, L., & Lanckriet, G. (2007b). Using games to collect semantic information about music. In *International Conference on Music Information Retrieval*. Vienna, Austria.

Tzanetakis, G. & Cook, P. (2002). Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, *10*(5), 293–302.

Tzanetakis, G., Ess, G., & Cook, P. (2001). Automatic musical genre classification of audio signals. In *Proceedings of the International Symposium on Music Information Retrieval*. Bloomington, IN, USA.

Urbano, J. (2011). Information retrieval meta-evaluation: Challenges and opportunities in the music domain. In *International Society for Music Information Retrieval Conference*, pp. 609–614. Miami, FL, USA.

Von Ahn, L. (2006). Games with a purpose. *Computer*, *39*(6), 92–94.

Von Ahn, L. & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 319–326. ACM.

Wack, N. (2011). Essentia & gaia: audio analysis and music matching c++ libraries developed by the music technology group. http://mtg.upf.edu/technologies/essentia. Last accessed Oct. 2011.

Westergren, T. (2010). *List of Music Genome Project attributes*. http://en.wikipedia.org/wiki/List_of_Music_Genome_Project_attributes. Last accessed Oct. 2010.

Whitman, B. (2005). *Learning the meaning of music.* Ph.D. thesis, Massachusetts Institute of Technology.

Whitman, B. & Ellis, D. (2004). Automatic record reviews. *Proceedings of the 2004 International Symposium on Music Information Retrieval.*

Whitman, B. & Lawrence, S. (2002). Inferring descriptions and similarity for music from community metadata. In *Proceedings of International Computer Music Conference*, pp. 591–598. Goteborg, Sweden.

Whitman, B. & Rifkin, R. (2002). Musical query-by-description as a multiclass learning problem. In *Proceedings of IEEE Multimedia Signal Processing Conference*, pp. 153–156. St. Thomas, USA.

Whitman, B., Roy, D., & Vercoe, B. (2003). Learning Word Meanings and Descriptive Parameter Spaces from Music. *Proceedings of the HLT-NAACL 2003 workshop on Learning word meaning from non-linguistic data-Volume 6*, pp. 92–99.

Witten, I. & Frank, E. (1999). *Data Mining: Practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann.

Wu, X., Zhang, L., & Yu, Y. (2006). Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pp. 417–426. New York, NY, USA: ACM.

Xu, R. & Wunsch, D. (2009). *Clustering.* Wiley-IEEE Press.

Yang, Y. & Chen, H. (2011). Ranking-based emotion recognition for music organization and retrieval. *Audio, Speech, and Language Processing, IEEE Transactions on*, (99), 762–774.

Yang, Y., Lin, Y., Su, Y., & Chen, H. (2008). A regression approach to music emotion recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, *16*(2), 448–457.

Zhao, Z., Wang, X., Xiang, Q., Sarroff, A., Li, Z., & Wang, Y. (2010). Large-scale music tag recommendation with explicit multiple attributes. In *Proceedings of the international conference on Multimedia*, pp. 401–410. Florence, Italy: ACM.

Appendix A: publications by the author

Journals

Sordo, M., Gouyon F., Sarmento L., & Celma Ò, Serra, X. (Submitted). Inferring Semantic Facets of a Music Folksonomy with Wikipedia. *Journal of Web Semantics.*
Marques, G., Langlois T., Gouyon F., Lopes M., & Sordo M. (2011). Short-term feature space and Music Genre Classfication. *Journal of New Music Research.* 40(2), 127-137.

Full-article contributions to peer-reviewed conferences

Sordo, M., Gouyon F., & Sarmento L. (2010). A Method for Obtaining Semantic Facets of Music Tags. *1st Workshop On Music Recommendation And Discovery (WOMRAD)*, ACM RecSys, 2010, Barcelona, Spain.

Marques, G., Lopes M., Sordo M., Langlois T., & Gouyon F. (2010). Additional Evidence That Common Low-level Features Of Individual Audio Frames Are Not Representative Of Music Genre. *7th Sound and Music Computing Conference*, 2010, Barcelona, Spain.

Herrera, P., Resa Z., & Sordo M. (2010). Rocking around the clock eight days a week: an exploration of temporal patterns of music listening. *1st Workshop On Music Recommendation And Discovery (WOMRAD)*, ACM RecSys, 2010, Barcelona, Spain.

Bischoff, K., Firan C., Paiu R., Nejdl W., Laurier C., & Sordo M. (2009). Music Mood and Theme Classification a Hybrid Approach. *Conference of the International Society for Music Information Retrieval (ISMIR).*

Laurier, C., Sordo M., Serrà J., & Herrera P. (2009). Music Mood Representations from Social Tags. *Conference of the International Society for Music Information Retrieval (ISMIR).* 381-386.

Serrà, J., Zanin M., Laurier C., & Sordo M. (2009). Unsupervised detection of cover song sets: accuracy improvement and original identification. *Conference of the International Society for Music Information Retrieval (ISMIR)* . 225-230.

Martínez, E., Celma Ò., Sordo M., De Jong B., & Serra X. (2009). Extending the folksonomies of freesound.org using content-based audio analysis. *6th Sound and Music Computing Conference.*

Sordo, M., Celma Ò., Blech M., & Guaus E. (2008). The Quest for Musical Genres: Do the Experts and the Wisdom of Crowds Agree?. *9th International Conference on Music Information Retrieval.*

Sordo, M., Laurier C., & Celma Ò. (2007). Annotating Music Collections How content-based similarity helps to propagate labels. *8th International Conference on Music Information Retrieval.*

Other contributions to conferences

Sordo, M., Celma Ò., & Bogdanov D. (2011). Audio Tag Classification using Weighted-Vote Nearest Neighbor Classification. *Music Information Retrieval Evaluation eXchange* (MIREX) extended abstract.

Sordo, M., Celma Ò., & Laurier C. (2009). QueryBag: Using Different Sources For Querying Large Music Collections. *Conference of the International Society for Music Information Research (ISMIR)*, Demo session.

Laurier, C., Sordo M., Bozzon A., Brambilla M., & Fraternali P. (2009). Pharos: An Audiovisual Search Platform using Music Information Retrieval Techniques. *Conference of the International Society for Music Information Research (ISMIR)*, Demo session.

Laurier, C., Sordo M., & Herrera P. (2009). Mood Cloud 2.0: Music Mood Browsing based on Social Networks. *International Society for Music Information Research Conference (ISMIR).*

Appendix B: supplementary material

Chapter 3: PCA covered variance

The following appendix presents additional material for the evaluations of the proposed music autotagging algorithm in Chapter 3. It includes a plot of the PCA components used in the aforementioned autotagging algorithm. Additionally, it depicts the list of the audio features which have the highest positive and negative contribution coefficient in the first and the second PCA components.

CAL500 dataset

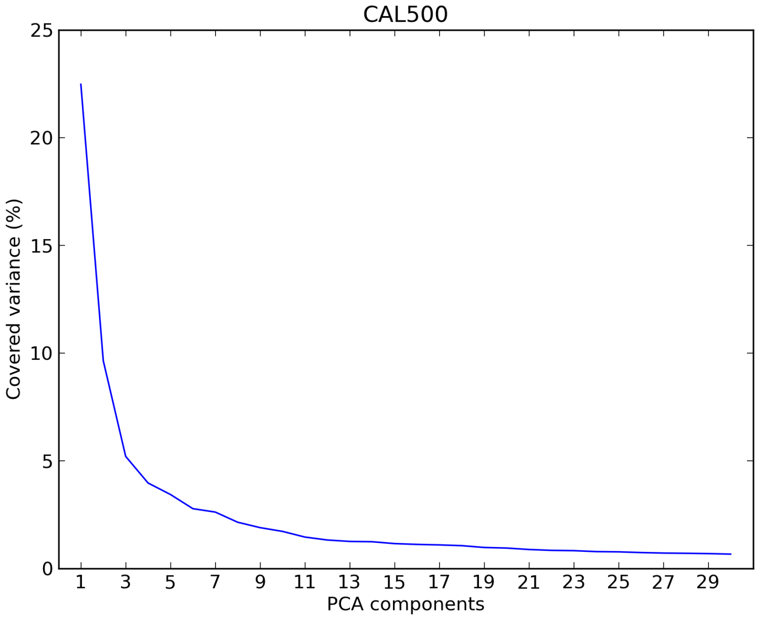

Figure 1: CAL500 dataset. Covered variance of each PCA component.

Table 1: CAL500 dataset. List of the 25 audio features which have the highest positive contribution coefficient in the **first** PCA component

Category	Feature	Value	Contrib.
High level	Mood (acoustic)	not acoustic	0.172
High level	Mood (sad)	not sad	0.171
High level	Mood (party)	party	0.162
High level	Mood (relaxed)	not relaxed	0.158
High level	Mood (aggressive)	aggressive	0.135
High level	Rhythm	fast	0.109
High level	Ballroom	jive	0.108
Low level	Pitch	dmean	0.107
High level	Genre (electronica)	dnb	0.105
Low level	Spectral centroid	mean	0.104
Low level	Zerocrossingrate	mean	0.103
Low level	Spectral rolloff	mean	0.102
Low level	Pitch	dvar	0.098
Low level	Spectral complexity	dmean	0.097
Low level	Pitch	var	0.093
Low level	Average loudness	–	0.091
High level	Genre (rosamerica)	roc	0.091
Low level	Barkbands spread	mean	0.085
High level	Mood (happy)	happy	0.081
Low level	Hfc	dmean	0.081
Low level	Spectral complexity	var	0.081
Low level	Spectral energyband middle high	mean	0.079
High level	Genre (dortmund)	rock	0.078
Low level	Spectral flux	mean	0.078
Low level	Hfc	mean	0.076

Table 2: CAL500 dataset. List of the 25 audio features which have the highest negative contribution coefficients in the **first** PCA component

Category	**Feature**	**Value**	**Contrib.**
High level	Mood (acoustic)	acoustic	−0.172
High level	Mood (sad)	sad	−0.171
High level	Mood (party)	not party	−0.162
High level	Mood (relaxed)	relaxed	−0.158
High level	Genre (electronica)	ambient	−0.136
High level	Mood (aggressive)	not aggressive	−0.135
High level	Mood	cluster3	−0.096
High level	Rhythm	slow	−0.091
High level	Mood (happy)	not happy	−0.081
High level	Timbre	bright	−0.069
High level	Ballroom	waltz	−0.067
High level	Gender	female	−0.065
High level	Genre (dortmund)	folkcountry	−0.064
High level	Voice/instrumental	instrumental	−0.063
Low level	Dissonance	dmean	−0.062
High level	Mood (electronic)	not electronic	−0.058
High level	Genre (rosamerica)	cla	−0.051
Low level	Pitch instantaneous confidence	mean	−0.050
High level	Genre (tzanetakis)	cou	−0.050
Low level	Spectral crest	var	−0.050
Low level	Silence rate 60dB	mean	−0.050
High level	Genre (dortmund)	jazz	−0.049
Low level	Dissonance	var	−0.048
Low level	Spectral crest	mean	−0.048
Temporal	First peak spread	–	−0.045

Table 3: CAL500 dataset. List of the 25 audio features which have the highest positive contribution coefficients in the **second** PCA component

Category	Feature	Value	Contrib.
High level	Mood (electronic)	not electronic	0.168
High level	Culture	western	0.152
High level	Genre (rosamerica)	roc	0.149
High level	Rhythm	fast	0.130
High level	Timbre	bright	0.123
High level	Genre (dortmund)	rock	0.112
High level	Mood (happy)	happy	0.111
High level	Genre (tzanetakis)	roc	0.101
Low level	Spectral energyband middle high	mean	0.097
High level	Ballroom	jive	0.095
High level	Gender	male	0.094
Low level	Spectral rms	mean	0.087
Low level	Spectral strongpeak	mean	0.086
Low level	Spectral energy	mean	0.083
Low level	Spectral complexity	mean	0.079
High level	Genre (dortmund)	folkcountry	0.076
High level	Mood (aggressive)	aggressive	0.075
Low level	Spectral strongpeak	dmean	0.070
Temporal	First peak spread	–	0.068
Low level	Hfc	mean	0.065
High level	Genre (electronica)	trance	0.062
High level	Genre (tzanetakis)	cou	0.062
High level	Mood (relaxed)	not relaxed	0.062
High level	Genre (dortmund)	alternative	0.061
Tonal	Hpcp [21]	mean	0.058

Table 4: CAL500 dataset. List of the 25 audio features which have the highest negative contribution coefficients in the **second** PCA component

Category	**Feature**	**Value**	**Contrib.**
High level	Mood (electronic)	electronic	−0.168
High level	Genre (dortmund)	raphiphop	−0.163
High level	Culture	non western	−0.152
High level	Timbre	dark	−0.123
High level	Rhythm	medium	−0.112
High level	Mood (happy)	not happy	−0.111
High level	Genre (electronica)	house	−0.104
Temporal	Beats loudness bass	dvar	−0.102
Low level	Spectral crest	dvar	−0.099
High level	Genre (rosamerica)	hip	−0.095
Low level	Spectral energyband low	dmean	−0.095
High level	Genre (tzanetakis)	hip	−0.095
Temporal	Beats loudness bass	var	−0.095
High level	Gender	female	−0.094
Low level	Pitch salience	dvar	−0.094
High level	Genre (tzanetakis)	reg	−0.091
Low level	Spectral rolloff	dvar	−0.091
Low level	Silence rate 60dB	mean	−0.090
Temporal	Beats loudness bass	mean	−0.089
Low level	Spectral rolloff	var	−0.086
Low level	Spectral rolloff	dmean	−0.085
Low level	Spectral flux	var	−0.081
Low level	Spectral rms	dmean	−0.081
Low level	Zerocrossingrate	var	−0.081
Low level	Spectral flux	dmean	−0.079

iTMS-500K dataset

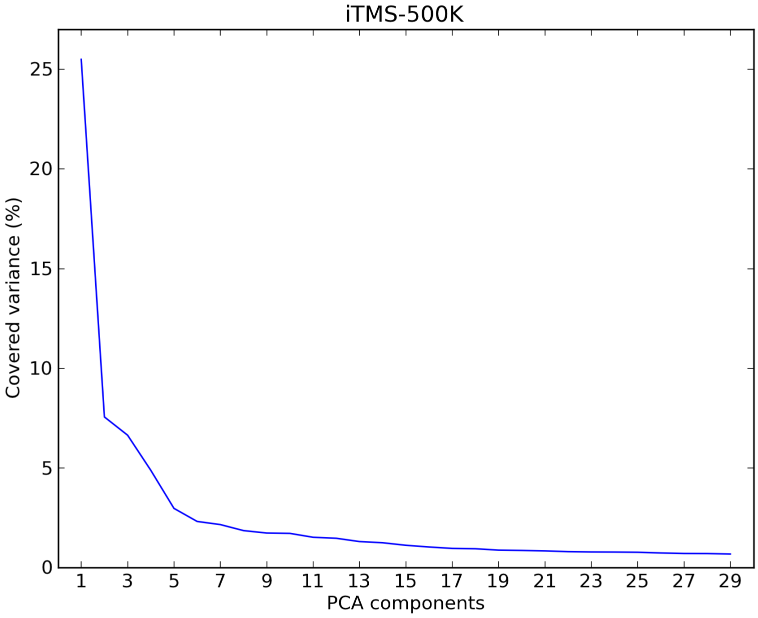

Figure 2: iTMS-500K dataset. Covered variance of each PCA component.

Table 5: iTMS-500K dataset. List of the 25 audio features which have the highest positive contribution coefficients in the **first** PCA component

Category	**Feature**	**Value**	**Contrib.**
High level	Mood (sad)	not sad	0.200
High level	Mood (acoustic)	not acoustic	0.192
High level	Mood (aggressive)	aggressive	0.188
High level	Mood (relaxed)	not relaxed	0.187
High level	Mood (party)	party	0.181
High level	Ballroom	jive	0.141
High level	Genre (rosamerica)	roc	0.133
High level	Rhythm	fast	0.130
High level	Genre (dortmund)	rock	0.122
High level	Genre (electronica)	dnb	0.109
Tonal	Thpcp [2]	–	0.093
High level	Mood (happy)	happy	0.092
Tonal	Thpcp [3]	–	0.088
High level	Gender	male	0.087
Tonal	Thpcp [32]	–	0.086
Tonal	Thpcp [4]	–	0.085
Tonal	Thpcp [34]	–	0.085
Tonal	Thpcp [17]	–	0.083
Tonal	Thpcp [33]	–	0.082
High level	Mood	cluster5	0.082
Low level	Barkbands spread	mean	0.081
Tonal	Thpcp [18]	–	0.080
Low level	Spectral flux	mean	0.078
Tonal	Thpcp [19]	–	0.077
Tonal	Thpcp [31]	–	0.077

Table 6: iTMS-500K dataset. List of the 25 audio features which have the highest negative contribution coefficients in the **first** PCA component

Category	Feature	Value	Contrib.
High level	Mood (sad)	sad	−0.200
High level	Mood (acoustic)	acoustic	−0.192
High level	Mood (aggressive)	not aggressive	−0.188
High level	Mood (relaxed)	relaxed	−0.187
High level	Mood (party)	not party	−0.181
High level	Genre (electronica)	ambient	−0.169
High level	Mood	cluster3	−0.132
High level	Rhythm	slow	−0.097
High level	Mood (happy)	not happy	−0.092
High level	Gender	female	−0.087
High level	Ballroom	waltz	−0.081
High level	Timbre	bright	−0.072
High level	Voice/instrumental	instrumental	−0.070
Low level	Silence rate 60dB	mean	−0.066
High level	Genre (dortmund)	jazz	−0.064
High level	Genre (rosamerica)	jaz	−0.061
Low level	Pitch instantaneous confidence	mean	−0.060
High level	Genre (dortmund)	folkcountry	−0.060
High level	Genre (rosamerica)	cla	−0.059
Low level	Spectral crest	mean	−0.053
High level	Genre (tzanetakis)	cla	−0.047
High level	Genre (rosamerica)	rhy	−0.047
High level	Ballroom	rumba-internat.	−0.044
Low level	Silence rate 60dB	var	−0.042
High level	Mood (electronic)	not electronic	−0.036

Table 7: iTMS-500K dataset. List of the 25 audio features which have the highest positive contribution coefficients in the **second** PCA component

Category	Feature	Value	Contrib.
High level	Voice/instrumental	instrumental	0.308
High level	Mood (happy)	not happy	0.259
High level	Timbre	dark	0.251
High level	Mood (electronic)	electronic	0.160
High level	Mood (aggressive)	aggressive	0.141
High level	Gender	female	0.105
High level	Mood	cluster5	0.089
High level	Rhythm	medium	0.086
High level	Genre (tzanetakis)	met	0.086
High level	Rhythm	slow	0.085
High level	Genre (rosamerica)	roc	0.084
High level	Genre (dortmund)	electronic	0.078
High level	Genre (tzanetakis)	cla	0.068
High level	Ballroom	waltz	0.068
High level	Mood (relaxed)	relaxed	0.062
High level	Culture	non western	0.055
High level	Genre (rosamerica)	cla	0.054
High level	Genre (dortmund)	jazz	0.054
High level	Mood (acoustic)	not acoustic	0.052
High level	Speech/music	speech	0.049
Tonal	Thpcp [3]	–	0.047
High level	Genre (rosamerica)	jaz	0.043
Low level	Pitch	mean	0.040
Tonal	Thpcp [2]	–	0.037
Tonal	Tuning equal tempered deviation	–	0.036

Table 8: iTMS-500K dataset. List of the 25 audio features which have the highest negative contribution coefficients in the **second** PCA component

Category	Feature	Value	Contrib.
High level	Voice/instrumental	voice	−0.308
High level	Mood (happy)	happy	−0.259
High level	Timbre	bright	−0.251
High level	Rhythm	fast	−0.171
High level	Mood (electronic)	not electronic	−0.160
High level	Genre (rosamerica)	pop	−0.146
High level	Mood (aggressive)	not aggressive	−0.141
High level	Genre (tzanetakis)	pop	−0.129
High level	Gender	male	−0.105
Low level	Silence rate 60dB	var	−0.104
High level	Mood	cluster2	−0.103
High level	Genre (rosamerica)	rhy	−0.083
High level	Genre (dortmund)	folkcountry	−0.081
High level	Ballroom	quickstep	−0.074
Tonal	Key strength	–	−0.070
High level	Genre (tzanetakis)	cou	−0.068
High level	Genre (dortmund)	pop	−0.066
High level	Mood (relaxed)	not relaxed	−0.062
Tonal	Thpcp [21]	–	−0.061
High level	Culture	western	−0.055
Tonal	Thpcp [15]	–	−0.055
High level	Mood	cluster4	−0.054
Tonal	Tuning diatonic strength	–	−0.054
Temporal	First peak weight	–	−0.052
High level	Mood (acoustic)	acoustic	−0.052

Chapter 5: extended results on assigning facets to tags

Table 9: List of the 76 top genres (from the top 103) produced by our system which are not present in the Gold Standard

Aida
Al_Green
American_Indian_music
Andreas_Scholl
Art
Bittersweet
Boy_soprano
brianmcknight
Cabaret
Cantautori
Chamber
Circus
Classical_composers
Contemporary
Contemporary_Christian_music
Contraltos
Countertenor
Countertenors
Dansband
Death
Drone
Electronic_dance_music
Ethereal_Wave
Ethnic
Featuring
Film
Filmi
Freak_folk
George_Michael
Gnawa
Gospel_blues
Indietronica
Indigenous
Indigenous_Australian_music
Islamic
Jeff_Buckley
Jesus
Levenslied
Lisa_Gerrard
Maritime
Mercedes_Sosa
Mezzo-soprano
Mezzo-sopranos
Military
Minimalism
Minimalist
Music_hall
Nico
Nusrat_Fateh_Ali_Khan
Oldies
Operetta
Political
Polka
Popular
Rautalanka
Religious
Rhythm_and_blues
Russian_chanson
Schlager
Selena
Side_project
Song
Soprano
Sopranos
Sufi
Symphonic_rock
Tenor
Tenors
Tin_Pan_Alley
Tosca
Traditional
VIA_music
Wedding
World
Zarzuela
Zolo

Table 10: List of the 71 top genres produced by the Baseline system (from a total of 103) which are not present in the Gold Standard

Act
African-American_Music
Anthem
Aria
Ballroom
Beat
Bebop
Black
Blue_Note
Boogie
Cantata
Chamber
Chant
Chorale
Christmas_Carol
Church
Comedy
Cool_Jazz
Country_and_Western
Drama
Ethnic
Folksong
Genre
Gospel
Gregorian_Chant
Hillbilly_Music
Hot_Jazz
Hymn
Landscape
Lead
Light_Opera
Macumba
March
Marching
Martial
Mass
Melodrama
Military
Modernism
Modern_Jazz
Movement
Neo_Jazz
New_Jazz
Operetta
Oratorio
Personal
Poetry
Popular
Postmodernism
Progressive_Rock
Prose
Psychedelicrock
Punk_Rock
Religious
Requiem
Rhythm_and_Blues
Rock'n'Roll
Scat
Scat_Singing
Scene
Serious
Sitcom
Skiffle
Sonata
Spiritual
Stream_of_Consciousness
Symphonic
Trad
Tragedy
Verse
Zydeco

Appendix C: definition of terms

The following is a list of some of the most used terms throughout this dissertation.

Cold start: a phenomena, usually prevalent in recommender systems, where a system cannot draw any inferences from users or items for which it has not yet gathered enough information.

Folksonomy: a system of classification derived from the practice of collaboratively and freely creating and managing tags, attached to any information resource.

Latent Semantic Analysis (LSA): a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that words appearing or not appearing in a document provide a set of mutual constraints, which determine the similarity of words or documents to each other.

Long tail: a term popularized by Chris Anderson (2006), it is a statistical property which informs that a larger share of population rests within the tail of a probability distribution, while a large mass is applied to a small subset of the population. Typically, this distribution is fitted into a power law distribution.

Popularity bias: when a system is more prone to propose / predict / recommend popular items more accurately, but ignoring or poorly performing in the rest.

Power law: a statistical relationship between two variables ($p(x) \propto x^{\alpha}$), where the frequency of an event varies as a power of some attribute of that event.

Semantic facet: a model describing a specific semantic concept of an item.

Semantic space: a mathematical representation of a large body of text. Every term or combination of terms has a high dimensional vector representation. In Latent Semantic Analysis, two terms are compared using the cosine of the angle between the vectors representing the terms. This occurs within a specific semantic space. A word cannot be directly compared between different semantic spaces.

Social tag: a tag introduced by users of any system to describe a content.

Tag: a keyword, category name, or meta data that describes any information resource.

Tag category: a group of tags with similar semantic meaning; related to semantic facets.

Tag vocabulary: a (structured or unstructured) set of tags.

Taxonomy: a system of classification arranged in a hierarchical structure or classification scheme, as defined by experts in the field.

Unbalanced dataset: a dataset that has many instances of some tags, but very few of others.

Weak labeling: a dataset is weakly labeled if the absence of a tag within a song does not necessarily mean that this song cannot be associated with that tag.