Algorithms and representations for supporting online music creation with large-scale audio databases

Gerard Roma Trepat

TESI DOCTORAL UPF / 2015

Director de la tesi

Prof. Dr. Xavier Serra Casals
Department of Information and Communication Technologies

To Anna

Acknowledgements

This thesis is obviously as much of a collective creation as the ones it hopes to enable. I have been very fortunate to meet and collaborate with a number of people during this journey. Thanks in the first place to Xavier Serra for the opportunity to work at the MTG, as well as for his inspiring vision and advice. I am also deeply indebted to Perfecto Herrera who has patiently guided and supervised my progress. Who knows where this would have ended without him holding the lamp. Very special thanks also for all this time to all the people who I have had the privilege to work with at the Phonondip, Metaverse and Freesound projects. In no particular order and surely with some omission Bram de Jong, Jordi Funollet, Frederic Font, Vincent Akkermans, Stelios Togias, Sergio Oramas, Alastair Porter, Martín Haro, Jordi Janer, Stefan Kersten, Mattia Schirosa and Nuno Hespanhol. Thanks also to Robin Laney, who allowed me to make a research stay at the Open University, as well as Chris Dobbyn and all the vibrant community I met there.

Thanks to everyone who I have met and chatted with at the MTG, also in random order, Graham Coleman, Nicolas Wack, Dmitry Bogdanov, Enric Aylon, Ricard Marxer, Tan Hakan Özaslan, Oscar Mayor, Ferdinand Fuhrman, Cyril Laurier, Hendrik Purwins, Justin Salamon, Mohamed Sordo, Piotr Holonowicz, Inês Salselas, Sergi Jordà, Carles F. Julià, Daniel Gallardo, Sebastian Mealla, Emilia Gómez, Enric Guaus, Jordi Bonada, Merlijn Blaauw, Marius Miron, Agustín Martorell, Juan José Bosch, Sankalp Gulati, Gopala Kuduri, Rafael Caro, Dara Dabiri, Georgi Dzhambazov, Ángel Faraldo, Cárthach Ó Nuanáin, Martin Hermant, Mathieu Bosi, Sertan Şentürk, Ajay Srinivasamurthy, Julio Carabias, Martí Umbert, Rafael

Ramírez, Julián Urbano, Álvaro Sarasúa, Zacharias Vamvakousis, Oriol Romaní, Hèctor Parra, Nadine Kroher, Panos Papiotis, Alfonso Pérez, Esteban Maestre, Cristina Garrido, Alba Rosado, Sònia Espí, Andres Lewin-Richter, and everyone I forgot. Also thanks to Lydia García for her endless patience. A special mention should go also to researchers from different places with whom I had the opportunity to collaborate: Waldo Nogueira, Massimiliano Zanin, Sergio Toral.

Finally, infinite thanks to Anna Xambó, who helped me in more ways than I would ever be able to write.

Abstract

The rapid adoption of Internet and web technologies has created an opportunity for making music collaboratively by sharing information online. However, current applications for online music making do not take advantage of the potential of shared information. The goal of this dissertation is to provide and evaluate algorithms and representations for interacting with large audio databases that facilitate music creation by online communities. This work has been developed in the context of Freesound, a large-scale, community-driven database of audio recordings shared under Creative Commons (CC) licenses. The diversity of sounds available through this kind of platform is unprecedented. At the same time, the unstructured nature of community-driven processes poses new challenges for indexing and retrieving information to support musical creativity. In this dissertation we propose and evaluate algorithms and representations for dealing with the main elements required by online music making applications based on large-scale audio databases: sound files, including time-varying and aggregate representations, taxonomies for retrieving sounds, music representations and community models. As a generic low-level representation for audio signals, we analyze the framework of cepstral coefficients, evaluating their performance with example classification tasks. We found that switching to more recent auditory filter such as gammatone filters improves, at large scales, on traditional representations based on the mel scale. We then consider common types of sounds for obtaining aggregated representations. We show that several time series analysis features computed from the cepstral coefficients complement traditional statistics for improved performance. For interacting with large databases of sounds, we propose a novel unsupervised algorithm that automatically generates taxonomical organizations based on the low-level signal representations. Based on user studies, we show that our approach can be used in place of traditional supervised classification

approaches for providing a lexicon of acoustic categories suitable for creative applications. Next, a computational representation is described for music based on audio samples. We demonstrate through a user experiment that it facilitates collaborative creation and supports computational analysis using the lexicons generated by sound taxonomies. Finally, we deal with representation and analysis of user communities. We propose a method for measuring collective creativity in audio sharing. By analyzing the activity of the Freesound community over a period of more than 5 years, we show that the proposed creativity measures can be significantly related to social structure characterized by network analysis.

Resumen

La rápida adopción de Internet y de las tecnologías web ha creado una oportunidad para hacer música colaborativa mediante el intercambio de información en línea. Sin embargo, las aplicaciones actuales para hacer música en línea no aprovechan el potencial de la información compartida. El objetivo de esta tesis es proporcionar y evaluar algoritmos y representaciones para interactuar con grandes bases de datos de audio que faciliten la creación de música por parte de comunidades virtuales. Este trabajo ha sido desarrollado en el contexto de Freesound, una base de datos de grabaciones sonoras compartidos bajo licencia Creative Commons (CC) a gran escala, impulsada por la comunidad de usuarios. La diversidad de sonidos disponibles a través de este tipo de plataforma no tiene precedentes. Al mismo tiempo, la naturaleza desestructurada de los procesos impulsados por comunidades plantea nuevos retos para la indexación y recuperación de información en apoyo de la creatividad musical. En esta tesis proponemos y evaluamos algoritmos y representaciones para tratar con los principales elementos requeridos por las aplicaciones de creación musical en línea basadas en bases de datos de audio a gran escala: archivos de sonido, incluyendo representaciones temporales y agregadas, taxonomías para buscar sonidos, representaciones musicales y modelos de comunidad. Como representación de bajo nivel genérica para señales de audio, se analiza el marco de los coeficientes *cepstrum*, evaluando su rendimiento en tareas de clasificación. Encontramos que el cambio a un filtro auditivo más reciente como los filtros de gammatonos mejora, a gran escala, respecto de las representaciones tradicionales basadas en la escala mel. Después consideramos tres tipos comunes de sonidos para la obtención de representaciones agregadas. Se demuestra

que varias funciones de análisis de series temporales calculadas a partir de los coeficientes *cepstrum* complementan las estadísticas tradicionales para un mejor rendimiento. Para interactuar con grandes bases de datos de sonidos, se propone un nuevo algoritmo no supervisado que genera automáticamente organizaciones taxonómicas basadas en las representaciones de señal de bajo nivel. En base a estudios con usuarios, mostramos que nuestro enfoque se puede utilizar en lugar de los sistemas tradicionales de clasificación supervisada para proporcionar un léxico de categorías acústicas adecuadas para aplicaciones creativas. A continuación, se describe una representación computacional para música creada a partir de muestras de audio. Demostramos, a través de un experimento con usuarios, que facilita la creación colaborativa y posibilita el análisis computacional usando los léxicos generados por las taxonomías de sonido. Finalmente, nos centramos en la representación y análisis de comunidades de usuarios. Proponemos un método para medir la creatividad colectiva en el intercambio de audio. Mediante un análisis de la actividad de la comunidad Freesound durante un periodo de más de 5 años, se muestra que las medidas propuestas de creatividad se pueden relacionar significativamente con la estructura social descrita mediante análisis de redes.

Resum

La ràpida adopció d'Internet i de les tecnologies web ha creat una oportunitat per fer música col·laborativa mitjançant l'intercanvi d'informació en línia. No obstant això, les aplicacions actuals per fer música en línia no aprofiten el potencial de la informació compartida. L'objectiu d'aquesta tesi és proporcionar i avaluar algorismes i representacions per a interactuar amb grans bases de dades d'àudio que facilitin la creació de música per part de comunitats virtuals. Aquest treball ha estat desenvolupat en el context de Freesound, una base de dades d'enregistraments sonors compartits sota llicència Creative Commons (CC) a gran escala, impulsada per la comunitat d'usuaris. La diversitat de sons disponibles a través d'aquest tipus de plataforma no té precedents. Alhora, la naturalesa desestructurada dels processos impulsats per comunitats planteja nous reptes per a la indexació i recuperació d'informació que dona suport a la creativitat musical. En aquesta tesi proposem i avaluem algorismes i representacions per tractar amb els principals elements requerits per les aplicacions de creació musical en línia basades en bases de dades d'àudio a gran escala: els arxius de so, incloent representacions temporals i agregades, taxonomies per a cercar sons, representacions musicals i models de comunitat. Com a representació de baix nivell genèrica per a senyals d'àudio, s'analitza el marc dels coeficients *cepstrum*, avaluant el seu rendiment en tasques de classificació d'exemple. Hem trobat que el canvi a un filtre auditiu més recent com els filtres de gammatons millora, a gran escala, respecte de les representacions tradicionals basades en l'escala mel. Després considerem tres tipus comuns de sons per a l'obtenció de representacions agregades. Es demostra que diverses funcions d'anàlisi de sèries temporals calculades a partir dels

coeficients *cepstrum* complementen les estadístiques tradicionals per a un millor rendiment. Per interactuar amb grans bases de dades de sons, es proposa un nou algorisme no supervisat que genera automàticament organitzacions taxonòmiques basades en les representacions de senyal de baix nivell. Em base a estudis amb usuaris, mostrem que el sistema proposat es pot utilitzar en lloc dels sistemes tradicionals de classificació supervisada per proporcionar un lèxic de categories acústiques adequades per a aplicacions creatives. A continuació, es descriu una representació computacional per a música creada a partir de mostres d'àudio. Demostrem a través d'un experiment amb usuaris que facilita la creació col·laborativa i dóna suport l'anàlisi computacional usant els lèxics generats per les taxonomies de so. Finalment, ens centrem en la representació i anàlisi de comunitats d'usuaris. Proposem un mètode per mesurar la creativitat col·lectiva en l'intercanvi d'àudio. Mitjançant l'anàlisi de l'activitat de la comunitat Freesound durant un període de més de 5 anys, es mostra que les mesures proposades de creativitat es poden relacionar significativament amb l'estructura social descrita mitjançant l'anàlisi de xarxes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Information and communication technologies keep rapidly transforming the way we create and enjoy music. Current usage patterns make it possible to support online collective creation of music based on large-scale audio databases. This thesis aims to provide the computational framework to model and support this practice, making it available to people with the most diverse backgrounds.

1.1 Motivation

Our approach is motivated first in the current context of change in the social habits and structures related with music creation and appreciation. Second, we analyze the continued shift towards networked technologies, which reflects in the increased support for connectivity in music creation tools. In this context, we consider the existing practice of sharing audio clips under Creative Commons (CC) licenses as an opportunity for supporting collaborative music creation based on information sharing.

1.1.1 Social context

During the last decades, the music business has changed dramatically. The recording industry had emerged from technical advances that allowed music to be recorded, packaged and distributed. These technological advances introduced many changes in how music was produced. For one, they changed the distribution of roles in music production, introducing new ones such as the producer or the engineer, and modifying the concept of authorship in

popular music. For another, they fostered the evolution of new forms of music that would not have been possible before, and radically changed existing ones. In the same way, the evolution and adoption of new technologies, mainly the generalization of Internet access, rendered the business of physical distribution of recorded music obsolete. While in the past cassette tapes allowed music listeners to duplicate and distribute music recordings, a turning point occurred when home duplication of digital CDs became cheaper (and then much cheaper) than the acquisition of commercial copies. Once most people connected to the Internet, the cost of duplication of digital music files became virtually zero. Thus, business models that try to continue with the idea of assigning a value to copies of the original recording are generally disappearing. At the same time, the renewed popularity of vinyl signals the loss in tangibility that music lovers have perceived in the new regime, which seems to indicate that the revolution has not finished.

It can be expected that these technology-driven changes will also result in a new distribution of roles in music creation, as well as in new forms of music. Given the large amount of people interested in music creation, the division between professionals and amateurs, which was mainly supported by the business model of the recording industry, has become blurry. This change can also be observed in other media, where participatory genres have become increasingly successful. Many television programs now involve and/or focus on common people rather than professional actors or journalists. In this context, it may be expected that, in a foreseeable future, music production and music consumption will be more intermingled. As the audience becomes more active and participatory, an opportunity exists for music that is no longer delivered as a finished product, but is created and appreciated collectively as it evolves. This idea can be seen as a (perhaps less solitary) reminiscence of Jaques Attali's utopian concept of *Composition*, as the stage (*network*) that follows *Repetition* (the period dominated by the recording industry):

> Finally, we can envision one last network, beyond exchange, in which music could be lived as composition, in other words, in which it would be performed for the musician's own enjoyment, as self-communication, with no other goal than his own pleasure, as something fundamentally outside all communication, as self-transcendence, a solitary, egotistical, noncommercial act. In this network, what is heard by others would be a by-product of what the composer or interpreter wrote or performed for the sake of

> hearing it, just as a book is never more than a by-product of what the writer wrote for the sake of writing it. (Attali, 1985)

1.1.2 Technological context

The idea of using Internet connections to enhance collaboration in music production is certainly not new. It has been tested in many ways both in engineering and artistic domains. During the 90s, Digital Audio Workstations (DAWs) took over the task of recording and producing music, and hard disks replaced analog and digital tapes. The interface of these DAWs combined the "piano roll" interface, that had been used for editing MIDI scores, with the multi-track digital audio editor inspired by tapes. Virtual instruments and effects could be added to each track, and an interface resembling hardware mixers was used for mixing tracks to the delivery format. This common environment allowed different levels of involvement with the possibilities of computer music, from the recording and mixing of acoustic instruments to music that was generated exclusively by computational means. However, computer workstations were designed for a single user. When Internet connections became available, attempts to add connectivity to DAWs were made, but failed to attract a significant user base. An example of it is the history of the company Rocket Networks[1].

As Internet has become more and more prevalent in most people's daily life, such attempts have re-emerged. Examples include the addition of networked collaboration to *Ableton Live*[2], a very popular DAW for audio-based creation, or the recently released *Ohm studio*[3]. One common feature of these programs is that they preserve the same interface and representation that served the single-user case. Social organization in collaborative DAWs is often limited to having tracks contributed by different authors. Also, some of the mentioned programs can render sections to non-editable clips that can be shared. However, the general idea of enforcing tracks limits the possibilities for sharing music fragments, for instance, by forcing that all effects and instruments are assigned to a given track for the whole project. Thus, the notion of tracks perpetuates the division of labour inherited from traditional instrument-based music, where each musician is expected to produce a sound stream. In this thesis, we propose a greater emphasis on supporting multiple hierarchical levels in music composition, allowing different levels of

[1] `http://www.jamwith.us/about_us/rocket_history.shtml`
[2] `http://www.ableton.com`
[3] `http://http://www.ohmstudio.com`

collaboration. Large-scale online information sharing and remixing (also known as "peer production" (Benkler, 2006)) is a relatively recent practice that has not been fully studied. Thus, our work does not focus on specific collaboration workflows, but on algorithms and representations that can be used to support many different models of collaborative music creation.

Another trend has been the development of web-based music creation. During many years, web browsers have been very limited with respect to audio and video, and relied on third-party plug-ins. The situation contrasts with the level of standardization of HTML and Javascript, which supported the rapid evolution of web application development. More recently, some functionality for low level audio processing (necessary for the development of music creation applications) was added to the Adobe Flash platform. A more comprehensive API for audio has been developed by the W3C standards body during the development of this thesis, and an implementation is already available in several web browsers (Adenot et al., 2013). These developments will allow leveraging the specific affordances of web applications that have become prevalent (e.g. relying on shared storage, as well as social and collaborative functionalities) for music creation.

Finally the potential of smartphones and tablets for music creation has generated great expectations. In these platforms, programs that rely on network services behave similarly to web applications: interaction is often determined by the rhythm of the HTTP requests and responses. Mobile apps take advantage of multi-touch screens and other sensors available in current smartphones. However, since it is generally simpler to sell client-only products, sharing is often limited to emailing your creations. In some cases, music creations can be shared through centralized storage, but the possibility of remixing existing compositions is rare. Due to the limitations in current mobile interfaces, mobile music applications are still regarded as mere toys by some users, but this situation is rapidly changing.

One important factor in the increased usage of Internet technology has been reliance on centralized storage. In recent times, the storage aspect of "cloud computing" has become mainstream and accepted to some extent by a large portion of computer (and especially smartphone and tablet) users. The opposition of centralized vs distributed (e.g. peer-to-peer) storage can be seen as a political one. This is especially true when centralized storage gives one party (typically a big company) the power to control and leverage the data. However, few people doubt about the need of centralized storage for many applications, especially for sharing data asynchronously. It would be

very difficult to imagine the world today without commodities such as web search engines. In this thesis, we focus on a specific model of centralized storage inspired by CC licenses, as implemented in Freesound. This site, created in 2005 in the context of the International Computer Music Conference (ICMC) has grown to become perhaps the most popular source of audio clips for music and media creators. Users from around the world upload recordings of the most diverse kind, along with textual descriptions that allow other users to easily find any sound they can imagine. Freesound was developed and keeps to be maintained at the Music Technology Group of Universitat Pompeu Fabra, where this thesis has been developed. The community of the site has always been at the center of any policy, and users retain the intellectual property of the content they upload (as per the CC license). It should be noted that many possibilities of social organization exist for centralized storage technologies. Some centralized resources such as web forums, emerge in a bottom-up fashion, and some achieve very large scales. Others are supported by start-up or established companies. Large audio repositories can also be used privately by music groups or collectives.

Our point of departure is then a large, shared database of audio clips. In this thesis we propose algorithms and data representations that allow exploring the possibilities of such resource for collective music creation. Perhaps the most useful developments that can be exploited for this purpose come from Information Retrieval, and particularly Audio Information Retrieval. However, the focus on music creation requires attention to many aspects that are not usually considered in this discipline. We will then use a multidisciplinary approach, with inputs from Human-Computer Interaction, Computer Music, and Computational Creativity research.

1.1.3 Audio as music material

The idea of using recordings as material for music creation was formulated by many artists and composers shortly after the popularization of the phonograph. As an example, hungarian artist Lázsló Moholy-Nagy speculated about producing music by direct inscription on wax discs as early as 1922 (Cox and Warner, 2004). Shortly after, in 1932, Rudolph Pfenninger had developed a technique for drawing sound waves on film (Levin, 2003). In 1936, Rudolph Amheim wrote:

> The rediscovery of the musicality of sound in noise and in language, and the reunification of music, noise and language in or-

> der to obtain a unity of material: that is one of the chief artistic tasks of radio (Battier, 2007).

The actual implementation of a music based on recordings is usually attributed to Pierre Schaeffer and the development of *Musique Concrète*. By cutting and splicing magnetic tape, the idea of making music out of existing recordings became practical. However, as demonstrated by Schaefer's writings, a systematic account of the new possibilities comparable to existing musical theory proved to be a complex issue (Chion, 1983).

Perhaps the most characteristic aspect of making music with existing recordings is that, unlike with other electronic means, such as synthesizers, one deals with a rigid medium, which tends to preserve its own acoustic qualities. At the same time, the recording affords the perception of a "unity of material", which allows thinking of music as a *discourse*, not unlike text or, nowadays, digital video[4]. The specific affordances and constraints of audio as a material have characterized many musical genres, such as *hip-hop* or *plunderphonics*. With the evolution of digital technologies, audio sampling has become just another tool in the palette of the electronic musician. However, the use of audio keeps signaling a certain change with respect to authorship and creativity. The different attitudes of musicians that face the authorship dilemma in the use of recorded audio (particularly when recorded by someone else or including sounds produced by someone else) could be classified in the following categories:

- Solipsism: the musician refuses to use any work that is not produced by herself. This attitude can be associated to a romantic view of creativity, where the artist is a creator that produces music out of nothing.
- Extreme transformations: the musician processes the samples up to the point that the original would be impossible to recognize. The result can be considered her own work by the choice of the transformations.
- Collage/citation: the use of recognizable samples with some degree of musical elaboration is accepted and the original author credited. This attitude is often associated with collage aesthetics.

[4]The idea of *Remix* as discourse has been recently investigated in the context of digital humanities (Navas, 2012)

- Anti-copyright activism: recognizable copyrighted music samples are used very explicitly as a way to criticize copyright law.
- Plagiarism: another author's work is used without credit or publicity, sometimes as a way to obtain an economic benefit.

In the end, the use of audio affords, perhaps more than other music creation techniques, a certain social perspective on musical creativity. A musician using a digital drum machine is effectively using a sample designed by someone else. This can extend to more sophisticated presets in digital instruments. Thus, the difference between using a digital instrument and using a sound downloaded from a website such as Freesound is not a difference in authorship but merely a licensing issue.

1.2 Aim of this thesis

The concept of "Remix Culture" has been proposed notably, among others, by Lawrence Lessig as a framework that permits and encourages derivative works. With the design of CC licenses, Lessig provided the tools for such culture to coexist with the current dominant culture, based on strict copyright laws. In *Remix* (Lessig, 2008), the concept was generalized to the idea of Read-Write (RW) culture, as opposed to Read-Only (RO) culture, and the world wide web, as originally designed by Tim Berners-Lee for sharing text documents, is described as an example of RW culture. Our aim is to help towards achieving a similar support for RW culture in the case of audio-based music.

Remix based on audio affords an inclusive framework, where very little background is necessary in order to start being creative. This contrasts with classical music training, where the student needs several years of training in order to be considered a valid interpreter of the work of established composers. While the democratization of composition may seem an aberration to some, this idea is hardly new, and can be easily related to evolution of popular music. The perspective depends probably (once more) on what one considers as "composition":

> In the Andaman Islands everyone composes songs, and children begin to practice themselves in the art of composition when they are still young. A man composes his as he cuts a canoe or a bow or as he paddles a canoe, singing it over softly to himself, until

> he is satisfied with it. He then awaits an opportunity to sing it in public, and for this he has to wait for a dance.... He sings his song, and if it is successful he repeats it several times, and thereafter it becomes a part of his repertory. If the song is not successful, the composer abandons it and does not repeat it. (Radcliffe-Brown, 1948) ap. (Attali, 1985)

Current practices in sharing recordings, such as in Freesound, provide an opportunity for online, audio-based, collective composition. An online community dedicated to composing and sharing music creations can be seen as a Community of Practice (Lave and Wenger, 1991; Johnson, 2001), where novices progress by learning from experts based on a shared interest.

Our aim is then to use and adapt techniques from information retrieval to the case of creative applications under this perspective. This includes different levels of musical elaboration, as well as user information. Particularly we distinguish three levels of information:

- Sound objects are samples that can be used as musical building blocks.
- Music fragments are, in the context of this thesis, groups of sound objects organized using some specific computational representation.
- Music creation communities are groups of users who create and share sounds and music fragments.

1.3 Thesis outline

The structure of this thesis follows the different kinds of information required for the aim of supporting online remix communities.

Chapter 2 reviews existing literature related with the aims of the thesis. First, it introduces the field of network music, with an emphasis on the web as a music creation platform. After an introduction to audio clip databases it summarizes relevant developments in audio retrieval that are relevant in order to leveragie these platforms for online music creation. Then it discusses existing approaches to music representation in computers, with an emphasis on grammars and support for nested structures. The chapter ends with a review of some models of collective creativity that have appeared in different domains.

In Chapter 3, we describe several approaches for indexing and retrieval of different kinds of audio samples. We deal specifically with the case of "unstructured" data, this is, audio data accumulated collectively without a predefined editorial process. We analyze representations based on analysis of the audio spectrum, and their summarization for dealing with the three main kinds of recordings that can be found in online databases: long field recordings, sound events and musical loops.

In Chapter 4, we describe both supervised and unsupervised machine learning approaches that can automatically produce taxonomical organizations of sounds with little or no textual information about what they contain. The described technologies can be used for leveraging large audio databases in online music creation.

In Chapter 5, we propose a representation for music artifacts as assemblages of audio samples, suitable for analysis and generation of music in web environments. We describe the framework of graph grammars as the foundation for this representation, and the algorithms that can be used for detecting frequent patterns.

In Chapter 6 we describe initial work towards designing for collective creativity in communities of practice based on sharing audio. We present a study based on the Freesound community that demonstrates the potential of complex network analysis for understanding the creative outcome of a community. We propose automatic measures derived from computational creativity models as guidance for the design and evaluation of software intended to support music creation communities, and show they can be predicted by analyzing their activities.

Chapter 7 concludes with some reflections about the potential and challenges involved in the proposed use case, and enumerates the novel contributions of this thesis.

Chapter 2

Background

2.1 Overview

The aim of supporting collective music creation on the basis of shared audio requires input from a number of disciplines. This chapter reviews prior work that is relevant for this purpose.

Online music creation can be seen as a form of *network music*, a field that has been investigated over the last decades. We start by introducing network music and reviewing prior work on web-based music creation.

We then focus on the potential of large audio databases originated from the established practice of sharing audio clips. The first requirement is then finding suitable sounds. We consider developments in content-based audio retrieval, which affords intuitive, non-verbal interaction with audio databases.

Storage and retrieval of musical artifacts based on those sounds should be based on some computational representation. We briefly review some representations relevant to our project that have been proposed in the context of computer music composition and musicological analysis.

Finally, we consider the challenge of designing systems for collective usage. Such systems require the specification of collective goals, as individual goals can be different and conflicting. We propose the notion of collective creativity, and describe some models that have been proposed for understanding such a complex process.

2.2 Network and web music

Network music is a small research area in computer music, which encompasses efforts to use computer networks in music creation. This work started as soon as it was feasible to set up a computer network on stage. Barbosa (2006) surveyed the field and proposed a classification space employed by the Computer-Supported Cooperative Work (CSCW) community to classify different works, according to the type of interaction (synchronous or asynchronous) and the spatial proximity (co-located vs remote interaction). Another classification was proposed by Weinberg (2005) depending on the level of interconnectivity. Rohrhuber (2007) distinguished two main modes of sharing information: *shared objects* vs *distributed objects*.

In practice, a broad distinction can be traced between networked music performance (potentially at different levels of improvisation) and networked creation. Most work has been done in the former problem, mostly from an artistic perspective, including co-located performers (e.g. laptop ensembles) and remote performers (often using general purpose video-conferencing software). Technically speaking, both synchronous and asynchronous interaction can be (and are) used in a performance.

In this thesis, we focus on the possibilities of the web as a platform for networked music creation. Web technologies afford concurrent interaction with large quantities of data in an asynchronous framework involving users from different locations and time zones. In this context, the border between performance and creation can be sometimes difficult to draw, particularly in projects where there is no more audience than the performers themselves. In practical terms, we can associate performance with quasi-synchronous interaction, when participants need to be using the system at the same absolute time. In the case of web-based environments this is necessarily an approximation, given the potentially long latencies and inherently asynchronous technologies. In turn, when participants are not required to interact at the same time, we can think of the system as a composition tool, allowing the user more time for thinking, revising and evolving a musical artifact before others have the chance to listen to it. However, because of the potential for conceptual continuity between both cases, it is worth to review previous work in both areas.

2.2.1 Web-based music performance

Web browsers are an atypical tool for music performance, but even more in the case of public performances. Many projects have explored performance on web environments, most of the time relaxing the requirement for strict synchronization, which often results in "jamming" style interfaces. Most of the systems are more oriented towards the collective experience of the performers than to public performances.

TransJam (Burk, 2000) is an early example of a web-based architecture. A generic http server allows multiple applications, each one is assigned a virtual room with chat capabilities. Applications make use of the *Jsyn* java library so they can be embedded in web pages as applets. An example application is *webdrum*, a drum sequencer that can be edited collaboratively in real-time. Also using transjam, *auracle* (Freeman et al., 2005) allows casual web users to control the music generation with their voice.

In DaisyPhone (Bryan-Kinns, 2004), a music loop is shared among participants who collectively add and remove notes with a choice of four different timbres. The initial project stressed HCI concepts such as *localization*, *mutual awareness*, *mutual modifiability* and *shared representation*. It has later been used to investigate *decay* (contributions of users that fade over time) (Bryan-Kinns and Healey, 2006) and *mutual engagement* (Bryan-Kinns, 2012).

In *Jam On* (Rosselet and Renaud, 2013) remote performers with no assumed music background interact asynchronously in real time by drawing lines on a canvas. Lines and their updates are distributed to HTML5 clients by a server.

2.2.2 Web-based music creation

While web-based performance usually relies on servers that support real-time messaging between clients, in web-based creation the role of the server is extended to storing the productions of users, which can then be re-used in collaborative workflows. Approaches to using the web as a platform for music creation can be drawn on in two main groups: on one hand, some composers have become interested in the web for involving web users in their own open compositions. Examples of this include William Duckworth's *Cathedral* (Duckworth, 2005), or Jason Freeman's *Graph Theory* (Freeman, 2008). Other projects have focused on analyzing the potential for collaboration in a more open setting. These are more related to the goal of this

thesis. However, the number of projects reported in an academic context is scarce.

FaustMusicOnline (Jordà and Wüst, 2001) was a pioneer project which explored collective composition over internet in 2001. An emphasis was put on allowing casual internet users and enforcing collaboration. The project was based on a native browser plug-in that allowed music creation by graphical interaction with a set of virtual vibrating strings. Compositions could be evolved from others, and a composition tree was used to store and browse the available compositions.

The CODES project (Miletto et al., 2005) proposed the concept of *Music Prototype*, an analogy of music with software prototyping that aims at facilitating music creation for people with different musical backgrounds through collaboration. Through the analogy with software development, several features that improve collaboration were imported, such as revision histories and logs. The system has been further used to understand how to support novices in music creation (Miletto et al., 2011).

CC-Remix and *Malleable Mobile music* (Tanaka et al., 2005) were two projects that explored the potential for bridging between music reception and social creativity using CC licenses. The first project was a web application that allowed a group of up to four users to engage in a collective remix session by choosing among previously segmented loops of a collection of CC-licensed songs. In the second project, the possibilities of portable devices for social interaction were explored, along with their sensors as sources for music control. Online chats were used to allow users forming groups. Content generation was also based on re-organization of segmented songs. The authors enumerate several HCI design concepts for analysis and evaluation of collective music creation: *shared goals*, *reciprocity*, *engagement*, *awareness* and *belonging*.

2.3 Freesound and audio clip sharing

Community-driven databases for sharing audio clips have become a resource with great potential for online music making. As an example, Freesound[1] now provides an HTTP API available to developers for accessing a large database of sounds, currently holding more than 200.000 audio files. The number of usable sounds could grow into much larger numbers if we consider

[1] `http://freesound.org`

existing methods for audio segmentation, such as the one we describe in chapter 3. On the other hand, the diversity that results from the activity of users around the world would be very difficult to achieve by other means. Other sites where one can find Creative Commons (CC) or royalty-free samples include Looperman[2] or Soundsnap[3]. Since the promotion of the CC licenses, sites like CC-mixter[4] or indaba-music[5] have focused on enabling collaboration mainly through sharing audio tracks. As we have mentioned, this can be seen as a direct affordance of the multi-track audio editor/ sequencer interface for traditional music workflows. However, the potential for applying existing research on content-based audio retrieval to this kind of resources remains largely unexplored.

2.4 Content-based audio retrieval

The concept or content-based audio retrieval can be seen as a general term encompassing different developments related with audio signals sharing a common framework. The objective is being able to find relevant audio files for a given task. To this aim, the audio signal is analyzed for obtaining meaningful representations. This process can be seen as "recognition" of high level concepts in the raw signal. In this sense, many of the developments in audio retrieval can be traced back to work on representation of speech signals, either for Automatic Speech Recognition (ASR) (Rabiner and Juang, 1993), speaker identification or even coding and transmission. Another field that has greatly developed audio recognition is content-based Music Information Retrieval (MIR) (Casey et al., 2008), usually dealing with polyphonic music signals available to consumers. On the other hand, indexing or recognition of environmental sounds can be seen from the point of view of Computational Auditory Scene Analysis (CASA) (Brown and Cooke, 1994). In the end, CASA can be seen as the most general case, as auditory scenes can contain any type of sound including environmental sounds, music and speech. In this thesis, we deal with the type of sounds that are most commonly shared for re-using, as opposed to files containing finished music or long speeches. This was the principle behind sites like Freesound, where finished songs are explicitly discouraged. The sounds in this site are still widely diverse, and a very general approach is necessary.

[2] `http://www.looperman.com`
[3] `http://www.soundsnap.com`
[4] `http://ccmixter.org`
[5] `http://www.indabamusic.com`

In this section, we analyze prior work dealing with three main classes of sounds, which can be associated with established practices in audio-based music creation. Long recordings are frequently used in *soundscape composition*. From a perceptual point of view, these recordings can be regarded as auditory scenes. Events occurring in such recordings, and all sorts of short recordings of any kind of sounds, can be related to the practice of sampling and *musique concrète*. Finally, it is very common to share music loops. Music loops can of course be seen as auditory scenes composed of many musical events, but we will consider them separately in order to account for their specific musical features.

2.4.1 Sound scenes

Since the popularization of flash-based digital recorders, obtaining reasonable quality recordings of environmental sounds has become easy and affordable. Thus, it is not surprising that long field recordings are among the most common type of audio files shared in sites like Freesound. From a music creation perspective, such recordings can be used in soundscape composition, a musical practice originated by Murray Schafer (Schafer, 1977) influenced by acoustic ecology. Soundscapes can be used for acousmatic music, but also in narrative or interactive audiovisual content such as video games, virtual environments, or movies. From a CASA perspective, such recordings would be defined as auditory scenes where several sound sources are usually combined into one perceptual stream or, sometimes, into a background and foreground. In recent times, research efforts on recognition of auditory scenes have grown, mostly inspired by potential applications in mobile platforms. Two tasks can be broadly identified across different applications and fields: one is the segmentation of long recordings into consistent scenes, and the other is the classification of scenes into distinct categories. Segmentation of long recordings into distinct scenes as been analyzed mainly in the context of movies and TV (Cai, 2005), and personal logs (Ellis and Lee, 2006). In this thesis, we focus on recordings that are shared by internet users, and assume a minimal manual segmentation to produce already consistent scenes.

The idea of classifying scenes into discrete categories has met growing interest with potential applications in robotics (Chu et al., 2006) and ubiquitous computing (Eronen et al., 2006). Most approaches can be classified in two groups: in the first case, some descriptors are computed from the audio signal using short overlapping windows. The most common are Mel

Frequency Cepstral Coefficients (MFCCs), which aim at capturing the spectral envelope in a compact way to approximate timbre sensations. Then, some statistical model is trained directly from the frame-level features. One helpful assumption is that scenes are mainly stationary signals, since their identification does not rely so much on long-term temporal evolution of audio features. Thus, in this case Gaussian Mixture Models (GMMs) are a common choice (Aucouturier et al., 2007) (Dargie, 2009). Low-order Hidden Markov Models (which become a GMM in the single-state case) have also been used (Eronen et al., 2006). The second approach consists in summarizing the entire recording into a single feature vector, and training the classification model with these vectors. Again, given the stationarity assumption it is common to use simple statistics of MFCC features. Support Vector Machines (SVM), K-Nearest neighbor (KNN) and GMMs are popular choices for the classification model (Chu et al., 2006). Other than traditional MFCC features, promising results (although very different depending on the class) have been obtained with matching-pursuit features (Chu et al., 2009), however, their computational cost may make them unpractical in the context of large audio databases. Methods based on vector-quantization of frame-level descriptors have also been tried in the context of consumer videos (Lee and Ellis, 2010).

One general problem in comparing different methods is the lack of common datasets and benchmarks. During the development of this thesis, the AASP *D-CASE* challenge (Giannoulis et al., 2013) was proposed and conducted for benchmarking different methods for recognition of auditory scenes and events. The baseline system proposed by the organisers was based on MFCCs and a frame-level GMM classifier, while systems with best performance used SVMs. The best score was achieved by our system (described in chapter 3) using a novel set of features for summarizing the evolution of MFCCs in intermediate windows of 400ms. Another approach achieved similar performance using a wavelet variant of MFCCs and other features averaged over windows of 4s, with an extra classifier to decide for the class of the whole recording (Geiger et al., 2013). Such intermediate "texture windows" are commonly used in MIR (Tzanetakis and Cook, 2002).

2.4.2 Sound Events

Zooming into scenes, we can decompose them into background and sound events. Sound is always the result of an interaction between objects or ma-

terials, and conveys information about some physical event (Gaver, 1993). Since the first experiments of Pierre Schaeffer, the idea of a "Sound Object" was central to the theory of *Musique Concrète*. However, it is difficult to find a clear definition in Schaeffer's work. At one point, the sound objects is simply defined as an "energetic event" (Schaeffer, 1966). From an acoustic ecology point of view, the distinction between background and events was proposed by Schafer (Schafer, 1977). The distinction has also been supported from experimental psychology (Guastavino, 2007). In CASA it is also customary to distinguish between auditory scenes and auditory events, the later being related with the theory of auditory streams (Bregman, 1994). Here, the problem is understanding how do we group simple elements resulting from the analysis of our hearing system, such as harmonic structures, clicks and modulations, so that they represent distinct events. This process is influenced by subjective learning, which makes it difficult to define precisely what is an auditory event. However, in practice short samples representing different "sounds" have been used for many applications. Intuitively, the idea of "a sound" can be related with a certain action performed on a resonant object, such as a musical instrument. Thus, sound events can often be recognized by a consistent spectral pattern and a time-varying energy envelope.

Like in the case of field recordings, two main tasks can be identified: detection of events and classification into discrete categories. In the case of sound events, both tasks can be carried on simultaneously. However, in this thesis, we will look at both tasks separately. The advantage is that this allows unsupervised indexing of segmented but unidentified events, i.e., finding consistent groups of sounds that can be used in the context of music creation, even if no labels are available. Joint detection and classification is usually done either using HMMs, or running a generic classifier such as an SVM over sliding windows (Xu et al., 2003).

The task of segmentation, this is, finding the location of sound events in longer recordings, has been approached from several disciplines. In speech recognition, effective techniques have been developed for Voice Activity Detection (VAD) (Ramirez et al., 2007), which have many applications in communication technologies. In MIR, a long tradition exists for the detection of either pitched or percussive events. From the perspective of environmental sound, systems have been developed for applications such as surveillance (Clavel et al., 2005), or indexing of video content (Xu et al., 2003). An important issue is whether the system can detect sound events that overlap in time. Literature on detection of overlapping events is scarce,

as this is a much more complex problem. In the context of this thesis, we assume non-overlapping events, on the basis that multiple simultaneous events (such as e.g. musical chords) can be usefully considered as individual sound objects for music creation.

The task of classification of sound events can be traced back to early experiments with audio indexing, mainly in the context of sound effects libraries. Initial works focused on small datasets assigned to a handful of concepts (Wold et al., 1996). Very high classification accuracies were obtained using HMMs (Zhang and Kuo, 1999). For larger scales, it is more common to aggregate features at the file level and use generic machine learning models. For example, statistics of a large set of features have been used along with K-NN classifiers for large scale applications (Cano et al., 2005). While SVMs are perhaps the most established generic classifiers, some concerns have been raised about their cost. For very large scales, this can be mitigated by using sparse features and linear kernels. Vector quantization can be a simple yet effective way to obtain such sparse features (Chechik et al., 2008) (Lee and Ellis, 2010).

Most of the described approaches are *supervised*, this is, aimed at recognizing pre-specified classes of events. Some work has been devoted to the problem of unsupervised discovery of short sound clips. In this case, most projects explored the use of Self-Organizing Maps (SOM) for clustering and presenting large collections of sounds. For example Pampalk et al. (2004) used a SOM for visualizing drum sample collections. An application to music files was described in (Pampalk et al., 2002). In (Brazil et al., 2002) a system for browsing sound effects using SOMs is described.

Finally, an important body of work has dealt with music creation using large databases of small audio fragments, in the tradition of corpus-based concatenative synthesis (Schwarz, 2007) and musical mosaicing (Zils and Pachet, 2001). These systems work by defining a target sound or score, or any data that can be described as a sequence of audio descriptors, and retrieve an optimal sequence to reconstruct the target from a database of sound segments. Segmentation is usually done either by fix intervals (usually short segments in the tradition of granular synthesis), score alignment or note onset detection. The problem of finding the optimal sequence of units (samples in the database) that match the target as closely as possible, is seen as a Constraint Satisfaction Problem (CSP). Although the original application was realistic instrument and voice synthesis, many approaches have been described for real-time operation (Schwarz et al., 2007). Ulti-

mately, the corpus-based paradigm affords a general perspective on the use of audio as a material for music (Schwarz, 2006).

2.4.3 Music loops

Music loops are very popular for music creation in many genres. During the 1990s a market emerged for CDs of audio loops. Currently, online stores for DJs such as Beatport[6] offer catalogs of loop packs. Community-supported sites like Looperman or Freesound also contain thousands of loops. Literature on indexing and retrieval of loops is, however, relatively scarce. Once more, we can distinguish between the task of identifying loops, and indexing them for retrieval.

With respect to identification of music loops, there has been work on the isolation of loops in polyphonic music (Ong and Streich, 2008). Here, frequently repeated patterns in music files are identified through the autocorrelation of chroma features. Detection of loops in polyphonic music, however, is a different problem from indexing databases of (already cut) loop files. The former is more related to music structure and to an interpretation of what constitutes a "relevant" loop in the context of repetitive music, although from a practical point of view, an infinite number of loops could be extracted from the same track. In this thesis, we deal with sound clips which may have been cut as loops or not. Our problem, then, is identify loopable material in the context of unstructured and possibly unlabeled data. This opens the possibility to find rhythmic patterns in all kinds of sounds, including environmental sound recordings.

Some works have explored indexing and retrieval of music loops. Audiocycle (Dupont et al., 2009) was a prototype that implemented analysis of musical loops, restricted to 4/4 rhythm meters and specific instrument classes. Although the dataset size is not explained, scaling to large databases is reported as future work. Other projects have focused specifically on drum loops, which affords a transcription approach (Gillet and Richard, 2004). Most of the mentioned works describe some kind of interface, either by using information visualization techniques to map the sounds to a 2D plane (Dupont et al., 2009; Ong and Streich, 2008) or using a Query-by-Example(QbE) approach (Kapur, 2004; Gillet and Richard, 2005). In general, research on indexing and retrieval of music loops lacks public datasets and evaluation metrics that facilitate comparison of different approaches.

[6] `http://www.beatport.com`

2.5 Music representation

The idea of music representation can be linked primitively to the need of memorizing music. In this sense, several cultures developed music notation systems in parallel to writing. These notations abstract and encode the music experience, and need to be interpreted by a performer. In a similar way, the phonograph and the magnetic tape provided more accurate ways of storing and reproducing music. Computers allow for many different ways of representing music.

In the 1990s, the issue of music representation on computers was extensively researched. Wiggins et al. (1993) surveyed the field while proposing an evaluation framework based on two dimensions: expressive completeness (where ultimately the most complete representation would be the waveform) and structural generality, which is better achieved by symbolic notations. Dannemberg (Dannenberg, 1993) surveyed a number of issues related with representation of different aspects of music. Here, waveform and symbolic representations such as scores are viewed as different abstraction levels, each one containing different information.

With the popularization of music programming languages, a general distinction can be made between "process-oriented" representations, i.e. when the computer program is itself a music piece, and "data-oriented" representations, when the musical information is encoded in some data structure that the program will use. In this thesis, we focus on on data-oriented representations, under the assumption that, as a natural fit for audio-based music they can lead to simple and easily understandable tools for casual web users that leverage the potential of large-scale audio databases.

2.5.1 Hierarchical representations and music grammars

Music often contains repetitions and hierarchical structures or groupings. Representation of these aspects is particularly required for enabling different participants to collaborate in the creation process. The simplest representation used for audio-based music creation, a list of time-stamped events, contains no hierarchy. Therefore, it stores no information on the different parts that may form a musical composition, including e.g. repeated parts or motifs. The *piano-roll* representation used in most audio sequencers can be considered as a collection of event lists. Thus, hierarchical information is limited to distinguishing between parallel tracks. A representation that

includes hierarchical information can be used for analyzing music pieces in addition to composition (Smaill et al., 1993).

Representation of hierarchical relationships is especially important for supporting collaborative creation, which requires division of labor and the exchange of parts of a musical composition. In this sense, many hierarchical representations have been proposed based on formal grammars. Formal grammars were introduced by Noam Chomsky in the 1950's as a means for formal analysis of natural language (Chomsky, 1957). The view of grammars as mathematical devices has since then remained at the foundation of formal languages and computer science. From the point of view of linguistics it was an important revolution that introduced a generative perspective: grammars served as language theories that could explain and predict linguistic phenomena by generating them from a formally defined set of laws. Chomsky classified grammars into four classes according to the level of restriction and generative power, from the most comprehensive type 0 (unrestricted), to the smallest subclass, type 3, which can parse only regular languages. Context-free grammars (type 2) are the basis of most programming languages, including those used to create music. In the 1970s and 1980s, the idea of modeling music with grammars became very popular. In web environments, traditionally with limited interaction capabilities but with access to shared data storage, analysis and generation can be used for computer-aided composition. Thus, reviewing these classic works can shed new light on the design of systems for representing sample-based music in web environments.

One of the first documented efforts to use formal grammars in music composition is due to researcher and composer Curtis Roads. In *Composing Grammars* (Roads, 1978) he described a system for music composition based on context-free grammars augmented with control procedures. The system provided the composer with a workflow for experimenting with structural and semantic aspects of composition. First, the composer would specify a grammar using a specialized language (*Tree*) and an associated compiler (*Gram*). The program would generate a compiler for the specified grammar. The composer would then work on valid derivations according to the grammar to create the "syntactic surface". A second language (*CoTree*), and its corresponding compiler (*GnGram*), would aid in the generation of the score. A final task, the "lexical mapping", consisted in pairing the terminals of the grammar with sound objects previously created by the composer. Such amount of relatively low-level tasks reflects the kind of interaction that computers supported at that time. Still, the use of *concrète* sound objects

and focus on nested structures makes this pioneering work relevant in the context of this thesis, so we will adopt part of this terminology.

In *Grammars as representations for music*, Roads and Wieneke (1979) presented a synthesis of formal grammar theory and surveyed the use of grammars for music analysis. Perhaps more importantly, they summarized the powers and limitations of the grammar approach. Considering iconic (i.e. based on analogies) and symbolic (based on convention) representations, it is quite obvious that, as symbolic representations, grammars rely on a discretization of the sound material. This limitation is however less restrictive for compositional purposes (where discrete symbols can be associated to arbitrary sounds) than for analysis of existing audio. A second limitation is the compromise in complexity, as the most complex types of grammars are often too complex to parse, while simple grammars can be too trivial and less effective than other models. A third limitation is that grammars are purely structural and hence they don't deal with the semantic and social implications of music. Despite these limitations, grammars have continued to be used in a number of computer music projects and studies, both for analysis and generation. Perhaps the best known is the system developed by Jackendoff and Lerdahl for the analysis of tonal music (Jackendoff and Lerdahl, 1981).

Holtzman's Generative Grammar Definition Language (GGDL) was developed as a tool for investigation of structural aspects of music using a computer (Holtzman, 1980). The language could be used both by composers and musicologists. GGDL supported "phrase structure rules" (standard string rewriting rules) and "transformational rules" (transformations such as transposition or inversion). It also provided a means for mapping abstract symbols to actual sounds synthesized with the possibilities offered by computers of the time. The system focused on unrestricted (type 0) grammars, and as a consequence it encouraged manual experimentation, offering limited automation capabilities.

Kippen and Bel's development of the Bol processor system (Bel and Kippen, 1992) has been extensively documented along different phases. The system was originally conceived for linguistic analysis of north-indian tabla music, where *bol* mnemonic syllables are used. Tabla music is usually improvised, typically involving permutations of a reference pattern. Expert musicians can assess whether a given permutation is correct or not. On this basis, the authors tailored several formal grammars that reflected correct variations. A second iteration of the Bol processor, named BP2 targeted

grammar-based music composition from a more general perspective, allowing composers to specify their grammars to generate compositions with arbitrary sound objects. Because of this focus on composition, BP2 omitted the parser mechanism and allowed a more free approach to grammar specification, subordinating the issue of correctness to aesthetic considerations.

Finally, one of the most well-known uses of grammars for music composition is David Cope's Experiments in Music Intelligence (EMI) (Cope, 2004). Over the years, Cope has refined a database-driven system that imitates the compositional style of classic composers. The works of the target composers are segmented and described in a database, and each fragment is assigned to a category according to a system called SPEAC: Statement, Preparation, Extension, Antecedent and Consequent. Such categories attempt to define a basic formalization of the dynamics of tension and relaxation in western tonal music. Thus, the system defines a set of rules that make a sequence of patterns of different categories correct. The music generation engine is based on an Augmented Transition Network (Woods, 1970), which allows for faster parsing and generation of context-sensitive rules.

In more recent times, many projects have explored the application of L-Systems (Lindenmayer, 1968) to music generation. L-Systems are formal grammars where the derivation is done in parallel (i.e. derivation rules are applied to a string of symbols simultaneously, so not taking into account their respective outputs). This feature was originally used to model the growth of algae, and has since been applied to the generation of realistic fractal images of different kinds of plants (Prusinkiewicz and Lindenmayer, 1996). Initial applications to music were based on these graphical interpretations, applied to scores (Prusinkiewicz, 1986). From there, most work has focused on how to interpret the strings produced by the algorithm to generate melodies or rhythms, either from pre-defined schemes (Worth and Stepney, 2005), interactive systems (McCormack, 1996) or evolutionary techniques (Lourenço et al., 2009) (Kaliakatsos-Papakostas et al., 2012).

2.5.2 Representations for the web

As more and more people got connected to the Internet, interest has grown on music representations that support the exchange of music information. After the success of HTML as the base language of the web, eXtensible Markup Language (XML) was developed as a more general and formal markup language for exchanging information between different programs and web services. Many XML representations have been proposed for

traditional western notation. Because of its hierarchical structure, music scores can be represented naturally in XML as hierarchies. Popular examples include MusicXML (Good, 2001) and Music Description Language (MDL) (Hewlett and Selfridge-Field, 2001). Still, there is a wide variety of musical practices that cannot be represented with western notation. An audio-oriented representation can be used in some cases. For example, the MPEG-7 standard (Salembier and Sikora, 2002) included the definition of the Segment Descriptor Scheme (DS), a description scheme for multimedia objects that allows defining them as compositions of segments. The MPEG-7 Segment DS allows the definition of hierarchies and arbitrary relationships between media segments such as audio. Still, the standard is more oriented to the description of existing content (e.g. for indexing and navigation) than to the creation of the new content. On the other hand, the standard addresses all kinds of multimedia content, including video, images and 3D. This generality adds a lot of overhead (which may be the cause that has precluded the standard from being generally adopted) and introduces several complexity layers that are not needed for music composition.

In the last few years, JavaScript Object Notation (JSON) has gained popularity as a simpler data representation for web applications, becoming the de-facto standard for JavaScript-centric applications. However, the use of JSON has been less prone to standardization than in the case of XML. As an example, MusicJSON, a JSON-based music representation has been proposed (Alvaro and Barros, 2012). Like MusicXML, MusicJSON focuses on traditional western music notation. Another trend has been the development of javascript-based music programming languages that take advantage of the web audio API (Roberts and Kuchera-Morin, 2012).

2.6 Models of creative communities

Applications for supporting collaborative work need to deal with collective objectives. The notion of collective creativity may be useful for modeling communities of online music creators and defining unified design criteria. Creativity is usually defined as the human ability to 'create' things that are new and have some sort of relevance. Beyond this definition there is generally little agreement. A general trend is to consider at least two levels (personal and historical creativity) at which novelty and relevance are evaluated. Personal creativity refers to ideas that are new and interesting to their author, while historical creativity refers to ideas or products that are considered innovative by a larger audience (Boden, 2003). While

the traditional discourse focuses on individuals and their mental processes, some researchers have emphasized the social dimension of creativity (Csikszentmihalyi, 1999) (Montuori and Purser, 1995). During the last decade, networks (mathematically formalized as graphs) have become widely used for formal analysis of social groups. Some researchers have analyzed collaboration and social networks in an attempt of gaining a better understanding of collective creativity.

A common hypothesis is that *small-world* network structures favor creativity and innovation (Cowan and Jonard, 2003). This idea has been tested empirically in different domains (Verspagen and Duysters, 2004) (Uzzi and Spiro, 2005) (Fleming et al., 2007). Small-world networks were formally defined by Watts and Strogatz (Watts and Strogatz, 1998), who characterized them as a crossover between a regular lattice (a graph where nodes are connected uniformly) and a random graph (where nodes are connected with a certain probability). Their model captured a property of many real world networks, where the average distance between any two nodes (computed from the *shortest path length*, i.e. the smallest number of edges connecting two given nodes) tends to be as low as in a random graph, while the clustering coefficient (in this case defined as the average of the fraction of nodes within the neighborhood of each node that are connected among them) is much higher. The general assumption when judging the small-world structure is that short path lengths benefit innovation by allowing the flow of information. The effects of high clustering are less obvious. Clustering is generally associated with the flow of redundant information, but also to grouping of similar agents which may improve collaboration (Cowan and Jonard, 2003). When analyzing the network of teams that produced Broadway musicals in the 50s, Uzzi and Spiro considered both short path lengths and clustering to impact creativity following a U-shape, in the sense that both properties are beneficial up to a certain point, from which they become detrimental due to homogenization (Uzzi and Spiro, 2005). Here, clustering is associated to the formation of conventions, and hence more related to the relevance than to the innovation component of creativity. A study based on collaboration networks in a patent database (Fleming et al., 2007) found no significant effect of clustering, and discussed the potential detrimental effects of clustering on innovation. A similar trend is observed for the interaction between high clustering and short path length, which is only shown to have an impact when clustering is relevant.

While coming from very disparate domains, studies of network creativity seem to coincide in that small-world network properties, especially short

path lengths, have an influence in the development of creativity. This idea can be used in the design of collaborative music creation applications based on networks of collaborators. We further develop this idea in chapter 6.

2.7 Open issues

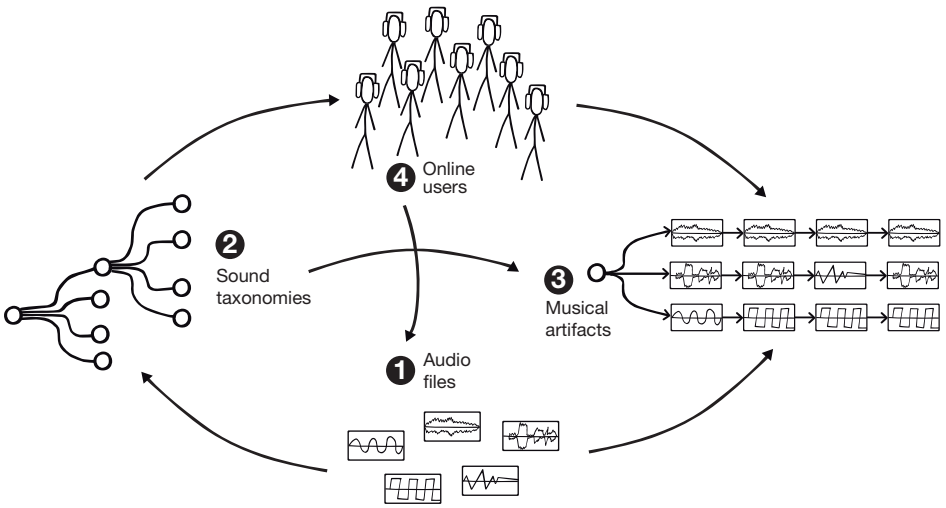


Figure 2.1: Elements involved in web-based music creation

In this chapter, we have reviewed existing research that is relevant for enabling the web as a medium for music creation based on shared audio data. We now recap the main issues involved in this use case and how we will deal with them in the following chapters (Figure 2.1).

The first problem is accessing audio databases. Content-based retrieval offers an opportunity for automatically indexing unstructured audio such as what we can currently find in sites like Freesound, without the need of manually labeling sound clips. In order to deal with general audio, this is, with any kind of recording that can be uploaded to a website, we consider three general classes of sounds: field recordings, sound events, and music loops. Also, in order to deal with this uncertainty about the type of uploaded audio, we follow a generic approach with separate steps for feature extraction and indexing. Chapter 3 covers strategies for automatic description of each of these three kinds of signals. In Chapter 4, we analyze generic strategies for indexing sounds based on these low-level descriptors.

A second problem is representing music during the creation process. In the case of music based on audio, this involves defining sequences and superpositions of audio clips. One open issue is how to represent different levels of music structure in order to allow sharing intermediate music products. We propose a suitable representation in Chapter 5.

Finally, designing applications for collaborative music making at a large scale requires representing and understanding user communities. We explore the notion of collective creativity as a global property that can be computed from data in existing audio sharing sites. In Chapter 6, we analyze how this property can be related to measures computed from user networks, which may in turn be determined by the design of web applications.

Chapter 3

Audio description in unstructured data

3.1 Introduction

In this chapter we explore the automatic extraction of content-based descriptors from unstructured audio data. In Chapter 2 we have reviewed content-based audio retrieval and its potential for audio-based music creation. By extracting low-level representations of audio signals, we can automatically index audio recordings and interact with them based on their acoustic properties, bypassing the need of manual annotation. This allows applications to integrate access to large audio repositories without breaking the creative flow.

Perhaps the main problem with this idea in the case of user-driven applications is the unstructured nature of the resulting database. Particularly in Freesound we can find almost any kind of recording that can be imagined. In order to deal with this diversity, we propose extracting a generic low-level representation that allows us to distinguish three main kinds of audio files: long field recordings, sound events and music loops. From the same low-level descriptor we can then adopt different strategies for obtaining summarized descriptions that can be used by indexing and retrieval algorithms.

Parts of this chapter have been published as (Roma et al., 2013)

3.2 Generic low-level description

Content-based audio retrieval relies on features computed from the sampled audio signal. Many features have been devised in the ASR, CASA, and MIR literature. Our aim is to work with completely unrestricted audio data, thus potentially containing music, speech, and environmental sounds. This means that a generic low-level representation is needed in order to learn at least what kind of sounds we are dealing with.

Probably the most common generic low-level representations are based on Mel Frequency Cepstral Coefficients (MFCC). MFCCs were initially defined by Mermelstein (1976) citing ideas from Bridle and Brown (1974) in the 1970s, in the context of ASR. However, they have also been extensively applied in the classification and clustering of both music (as a representation of monophonic or polyphonic timbre) and environmental sounds. MFCCs were based on the Mel scale, a logarithmic map of frequency defined in 1937 based on psychoacoustic experiments of pitch perception. While MFCCs have become standard in many disciplines, it is common to find low-level features inspired in more recent frequency scales. From a signal processing point of view, many of these proposals can be considered in the same framework defined by MFCCs. In this framework, a frequency domain, frame-level representation is first obtained from the Short-time Fourier Transform (STFT). This implies slicing the audio waveform into overlapping segments of equal length, to which a windowing function is applied for smoothing the boundary discontinuities. The length of the segment determines a trade-off between time and frequency resolution. The magnitude spectrum obtained from the Fourier transform of the windowed segment (usually using the Fast Fourier Transform, FFT) is then quantized by passing it through a bank of band-pass filters, with center frequencies distributed according to some perceptual scale. Applying the filter bank requires merely a multiplication in the frequency domain. The result can already be used as an audio feature, which can be easily interpreted visually as a spectrogram. In order to improve the performance of further computations, Mermelstein introduced a compression step, by computing the log of the filtered magnitudes. Finally, the Discrete Cosine Transform (DCT) is computed in order to de-correlate the log magnitudes. The result is defined as the *cepstrum*, which has the property of telling apart fast oscillations in the spectrum, usually associated with pitch in the case of voiced sounds, from its envelope associated with timbre. Thus, the first DCT coefficient represents energy (DC of the spectrum), and successive ones represent progressively faster oscillations.

A common configuration is taking the first 13 coefficients of a 40 bands filterbank. This representation of the spectral shape gives a very generic and efficient description of audio. The de-correlated features allow more accurate computation of distance measures, which are needed for most tasks related with indexing and retrieval. Several descriptors can be described under the MFCC framework depending on the frequency scale and shape of the filters, and whether the raw filterbank or cepstral features are used. We now describe three commonly used approaches.

3.2.1 Mel Frequency Cepstral Coefficients

MFCCs were developed in Mermelstein's original framework. The scale used for the center frequencies of the filterbank is the Mel scale, which was developed from experiments involving judgments of distance among pitched tones. These follow a logarithmic curve. While several formulas have been proposed to approximate this scale, one of the most commonly used is:

$$f_{mel} = 1127 log(\frac{1+f}{700}) \tag{3.1}$$

The filters follow a triangular shape with centers uniformly spaced along the mel scale, reaching the max in its own center frequency and ending in the next one. Hence, in theory, bandwidth depends on the number of filters, although it can also modified by some factor. The triangles are calculated to have either constant height or constant area. All of this can be trivially implemented from the result of the DFT. One advantage of MFCCs is that a plethora of implementations is available for any platform. At the same time, these introduce many details and parameters that may have an impact depending on the task at hand. For example, the *rastamat* matlab package (Ellis, 2005) used in our experiments can be used to reproduce the results of several implementations. A comparative evaluation for the task of speaker identification found small variations in results (Ganchev, 2005). As we have seen in Chapter 2, MFCCs are probably the most ubiquitous features in speech, music and environmental sound recognition.

3.2.2 Bark bands

Another frequency scale, the Bark scale, was proposed by Zwicker, after the initial experiment by Fletcher defining critical bands of hearing (Fastl and Zwicker, 2001). In short, such experiments determined the bandwidth at which a noise signal is able to mask a pure tone at a given frequency.

This gives an intuition of the resolution limits in frequency selectivity of the cochlea, the "critical band", from which the notion of a bank of "auditory filters" can be derived. Zwicker initially published a table with the measured frequency bands, which has allowed direct implementations by quantizing the magnitude spectrum according to these bands. However, he stressed that the relevant fact in those numbers was the bandwidth, and not the specific frequencies of thefilters, as the hearing system is able to adapt the central frequencies to incoming sounds. Hence, we can use the bark scale in the same framework as MFCCs. Simple formulas for the bark scale have also been poposed (Traunmüller, 1990):

$$f_{bark} = 13arctan(0.00076f) + 3.5arctan((\frac{f}{7500})^2) \tag{3.2}$$

The filter shape is usually a rectangular or trapezoidal shape in the logarithmic (dB) space. The bandwidth should thus be given by the distance between two whole barks. Bark bands, as raw features based on the bark scale and rectangular filters have been used in many occasions for audio indexing and retrieval, especially for creative applications (Herrera et al., 2003; Pampalk et al., 2004; Jehan, 2005). The advantage of using raw bands is that their values are easy to understand and relate with sound perception. They have also been used for obtaining cepstral coefficients, or considered as an alternative filterbank for MFCCs (Shannon and Paliwal, 2003; Ellis, 2005).

3.2.3 Frequency domain ERB gammatone filterbank

The Equivalent Rectangular Bandwidth (ERB) scale was presented by Moore and Glasberg (1996) as an improvement over the Bark scale devised by Zwicker, which they attributed to a measurement error. The underlying model, developed mainly by Patterson et al. (1992), offered a better explanation on the relationship between loudness, frequency and masking in the auditory system. The filter bandwidth can be expressed as a function of frequency (Moore, 2012):

$$ERB = 24.7(\frac{4.37f}{1000} + 1) \tag{3.3}$$

Here, the constant term can be interpreted as the minimum bandwidth ($minBW$), and the term multiplying the central frequency, as the reciprocal

of the auditory filter's Q ($EarQ$) (Slaney, 1993):

$$ERB = \frac{f}{EarQ} + minBW \tag{3.4}$$

A frequency mapping scale analogous to the mel and bark scales can be also derived:

$$f_{ERB} = 21.4log10(\frac{4.37f}{1000} + 1) \tag{3.5}$$

Patterson's model also included the *gammatone filter* of nth order, with impulse response

$$gt(t) = a^t(n-1)e^{(-2\pi bt)}cos(2\pi ft + \Phi) \tag{3.6}$$

for a given carrier signal with frequency f, phase ϕ and amplitude a, where b is the bandwidth of the filter. This is generally considered to be a better approximation of the auditory filter shape. Features based in the time-domain gammatone filterbank have been used in several works, e.g. (McKinney and Breebaart, 2003). On the other hand, gammatone filters can also be approximated by multiplication in the frequency domain (Ellis, 2009). This approximation can be fit in the MFCC framework to provide a more accurate version with respect to the psychoacoustic model, while providing a computationally cheaper representation than time-domain filters. Such features, which can be named *GFCC* have also been used in speech recognition (Shao and Wang, 2008)(here, the authors use the acronym for Gammatone Frequency Cepstral Coefficients, but Gammatone Filterbank Cepstral Coefficients would equally work).

Figures 3.1, 3.2 and 3.3 show the magnitude response of the filter bank for each of the three commonly used features. In this thesis, we adopt GFCC as a generic low-level audio representation. Thus, from here on, we use G to denote the sequence of spectral frames $G_1, G_2, G_3...$ that make the frequency domain gammatone spectrogram, and G_c to denote the sequence of cepstral coefficients derived from G.

3.3 Feature aggregation

One general problem for content-based indexing of audio files is feature aggregation. Most common set-ups require the integration of frame-level

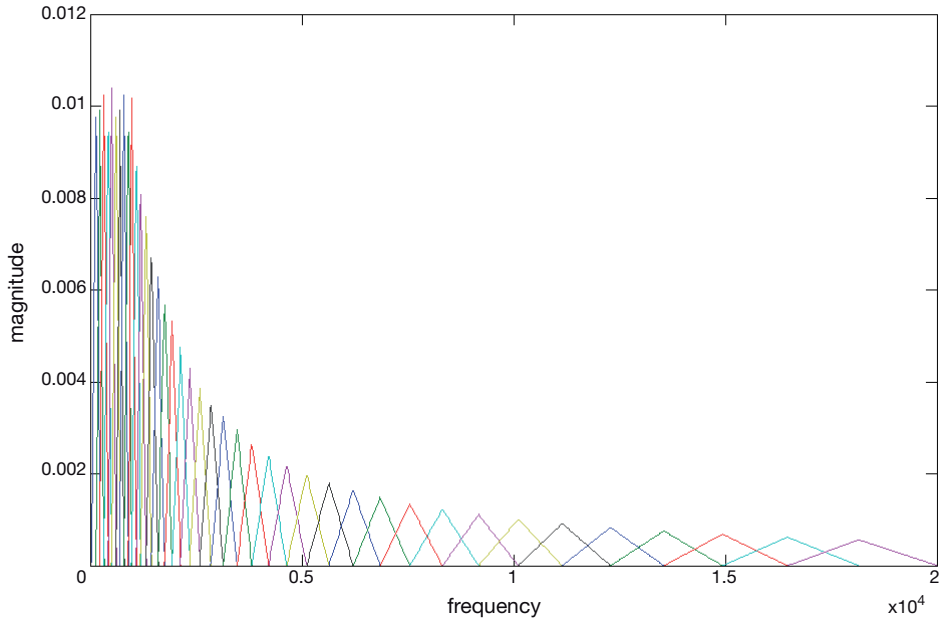


Figure 3.1: Mel scale filterbank (with constant area)

features over some period of time in order to obtain a single vector that can be fed in to algorithms for classification and clustering. The typical approach consists on averaging the frame-level features, a process in which information about the temporal evolution and distribution of the features is lost. The development of features that describe the temporal evolution of the sound is still an open issue. In this thesis, we adopt a different perspective for the three general classes of sounds mentioned in Chapter 2. It is clear that summarizing the temporal evolution of frame-level features must be done differently for field recordings, sound events, and music loops.

In the case of field recordings, summarization can be done statistically, as the signal may be considered stationary. Field recordings may contain different events, but the identification of the context, typically associated with locations (a busy street, a park, a beach ...) is done in the long term. Contrastingly, recognition of events can be done by analyzing the temporal evolution of the frame-level features, for example in the energy envelope. Finally, summarizing music loops will be more useful if the regularities in the evolution of the features are taken into account. In order to apply different summarization strategies to each kind of sound, it will be necessary

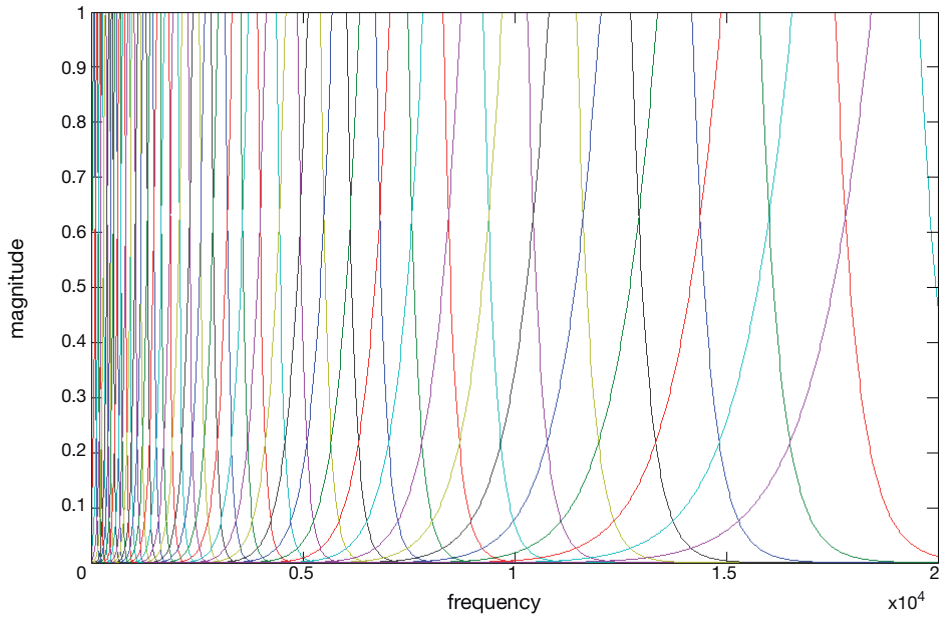


Figure 3.2: Bark scale filterbank

to identify them first. We now describe our strategies for recognizing and then summarizing each type of sound.

Sound scenes

Recognition Field recordings are the longest and most diverse type of recordings that we consider in this thesis. This type of recording is traditionally performed in locations where interesting sounds can be captured. Traditionally, the recording equipment stays in the same place for all of the recording. However nowadays it easy to make continuous recordings where the equipment (e.g. a mobile phone) moves around locations. From a CASA point of view, an ensemble of sounds recorded in a given location can be regarded as an *auditory scene*. However, in order to refer to the recording (and not to its perception), we will refer to *sound scenes*. Recognition of different sound scenes in the same recording has been considered in the context of TV audio or personal logs, as described in chapter 2. In the first case, a movie or a long video recording of TV programs can contain different scenes. In the second, a recording of a person moving through the

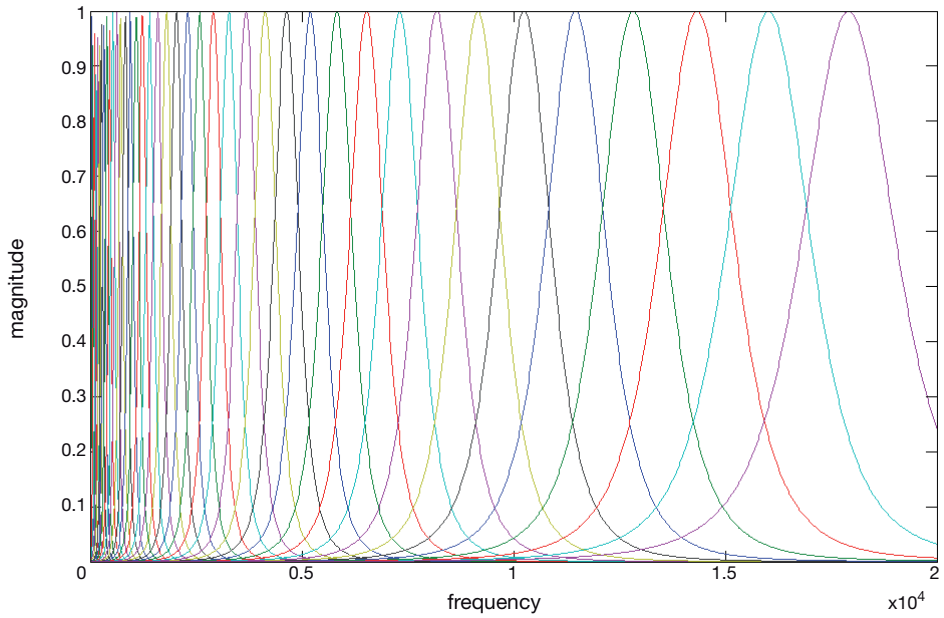


Figure 3.3: ERB scale gammatone filterbank

day with a portable recorder may contain audio from different locations. Here, we analyze sounds that users contribute to a database because they regard them as reusable material for music or audiovisual creation. Thus, we assume that in the worst case they will contain a consistent sound scene.

Telling apart long environmental recordings from other sounds in an unstructured database can be done simply based on heuristics. We take Freesound as a case in point. Since finished songs are not allowed in the site, it is unlikely that long recordings will contain music (note that even so, music could be considered a type of sound scene, and methods for indexing them based on statistics would still work in some way, although critically failing to take into account music features). Also, recordings of a single sound event will be generally limited in time by the ability of the resonating object to preserve energy. Thus, duration can already be a valid heuristic. To illustrate this, we may consider the *field-recording* tag in Freesound. Figure 3.4 shows the probability that a sound contains this tag as a function of its duration (quantized to seconds). This probability quickly grows for files lasting more than a few seconds. Considering that tags are freely assigned, even the probabilities of at least 30% are a high value ("field recording" is

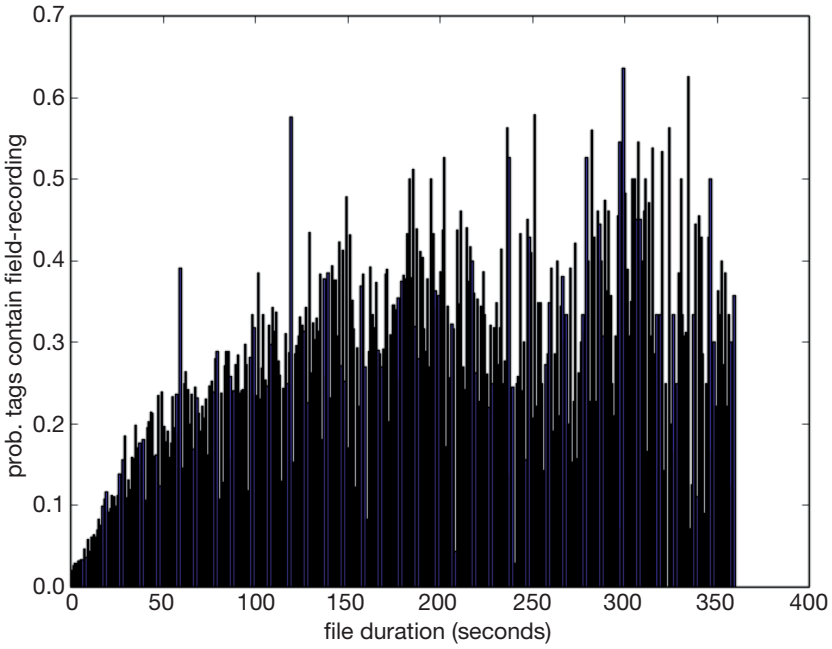


Figure 3.4: Duration of sounds tagged as "field-recording" in Freesound

actually the most common tag in the site). Another heuristic can be simply the number of detected events contained in a recording. By using some event detection algorithm (such as the ones described later), the number of events can also be used to decide if a sound is treated as a sound scene. In the end, all recordings can be related to a sound scene of some sort, so the decision is mainly pragmatic.

Aggregation In the case of sound scenes, aggregation can be based on statistics. While the scene can contain several events, we assume that human recognition of the secene will be based on long term statistics. However, we can still consider the general "texture", and characterize its short-term variations. For example, traffic noise may be characterized by short-term periodicities of car engines, while a background where wind or water dominate may characterized as filtered noise. We propose Recurrence Quantification Analysis (RQA) (Zbilut and Webber, 2006) to characterize these short term temporal dynamics as an addition to traditional statistics. RQA

is a set of techniques developed during the last decade in the study of chaos and complex systems. The basic idea is to quantify patterns that emerge in recurrence plots. RQA has been applied in a wide variety of disciplines, although applications to audio recognition are scarce. To the best of our knowledge there has been no prior work using RQA on time series of spectral or cepstral coefficients. RQA features derived from frame-level chroma features have been tested in the cross-recurrence setting, where two different series are compared, for cover song detection (Serrà et al., 2009). The original technique starts from one-dimensional time series which are assumed to result from a process involving several variables. This multidimensionality is recovered by delaying the time series and embedding it in a phase space. The distance matrix of the series is then computed and thresholded to a certain radius r. The radius represents the maximum distance of two observations of the series that will still be considered as belonging to the same state of the system. In our case, we already have a multivariate signal to represent the audio spectrum via cepstral coefficients. Hence, we adapt the technique by computing and thresholding the distance matrix obtained from the GFCC representation using cosine distance. Thus, if we denote the series of feature vectors as the multivariate time series G_c of length N as $G_c = G_{c1}, G_{c2}, G_{c3}...G_{cN}$, then the recurrence plot R is defined as

$$R_{i,j} = \begin{cases} 1 & \text{if } (1 - \frac{G_{ci} \cdot G_{cj}}{||G_{ci}||||G_{cj}||}) < r \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

Figure 3.5 shows the different steps of the process from the gammatone spectrogram up to the recurrence plot. The main intuition is that diagonal lines represent periodicities in the signal, i.e. repeated (or quasi-repeated, depending on the chosen radius) sequences of frames, while vertical lines (or horizontal, since the plot is symmetric) represent stationarities, i.e. the system remains in the same state. From this idea, several metrics have been developed that quantify the amount and length of lines of contiguous points in the matrix. Most features were developed by Webber and Zbilut (1994). We extract the most commonly used ones and add some more variables in order to obtain more features for classification and clustering.

- Recurrence rate (REC) is just the percentage of points in the recurrence plot.

$$REC = (1/N^2) \sum_{i,j=1}^{N} R_{i,j} \tag{3.8}$$

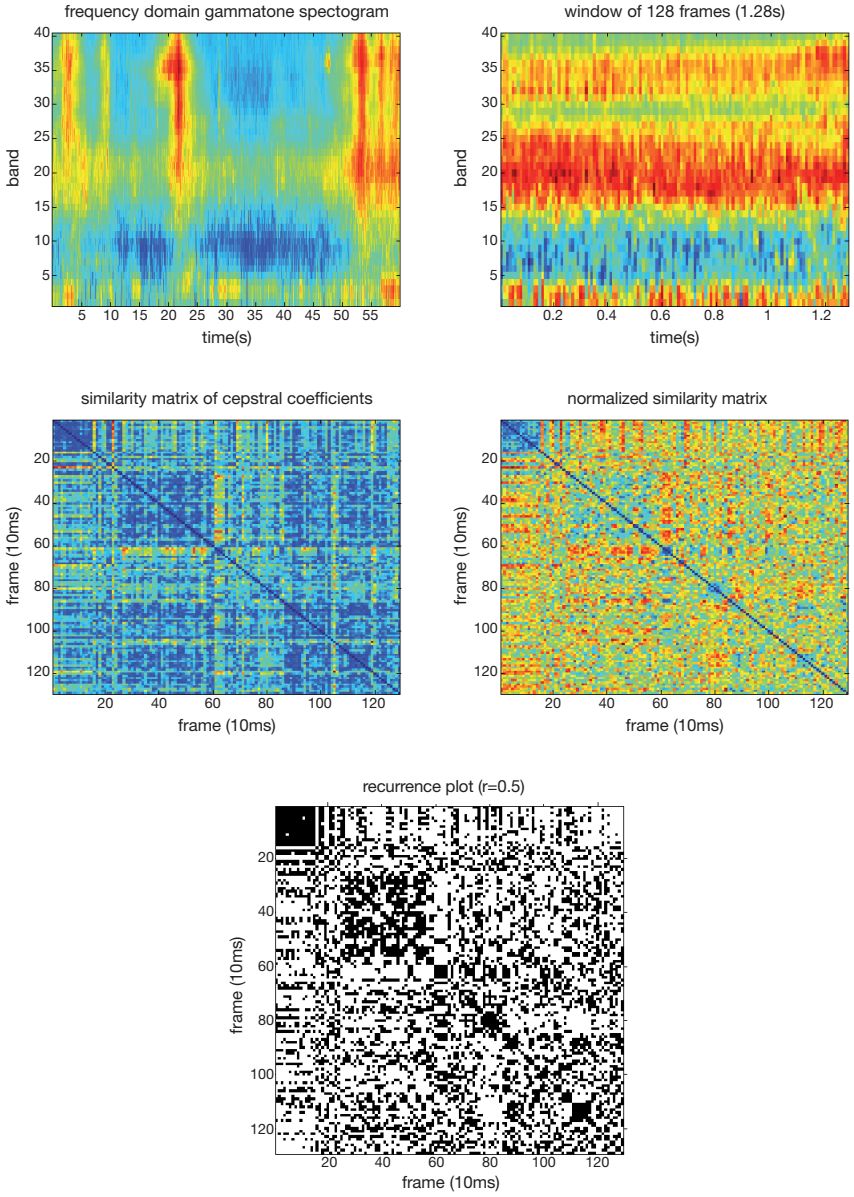


Figure 3.5: Steps in the construction of the recurrence plot

- Determinism (DET) is measured as the percentage of points that are in diagonal lines.

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}} \tag{3.9}$$

  where $P(l)$ is the histogram of diagonal line lengths l

- Laminarity (LAM) is the percentage of points that form vertical lines.

$$LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=1}^{N} vP(v)} \tag{3.10}$$

  where $P(v)$ is the histogram of vertical line lengths v

- The ratio between DET and REC is often used. We also use the ratio between LAM and REC, so we define them as

$$DRATIO = N^2 \frac{\sum_{l=l_{min}}^{N} lP(l)}{(\sum_{l=1}^{N} lP(l))^2} \tag{3.11}$$

$$VRATIO = N^2 \frac{\sum_{v=v_{min}}^{N} vP(v)}{(\sum_{v=1}^{N} vP(v))^2} \tag{3.12}$$

- LEN and Trapping Time TT are the average diagonal and vertical line lengths

$$LEN = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=l_{min}}^{N} P(l)} \tag{3.13}$$

$$TT = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=v_{min}}^{N} P(v)} \tag{3.14}$$

- Another common feature is the length of the longest diagonal and vertical lines. The inverse of the maximum diagonal (called Divergence) is also used. We use the inverse of both vertical and diagonal maximum lengths

$$DDIV = \frac{1}{max(l)} \tag{3.15}$$

$$VDIV = \frac{1}{max(v)} \tag{3.16}$$

- Finally, the Shannon entropy of the diagonal line lengths is commonly used. We also compute the entropy for vertical line lengths.

$$DENT = -\sum_{l=l_{min}}^{N} P(l)ln(P(l)) \tag{3.17}$$

$$VENT = -\sum_{v=v_{min}}^{N} P(v)ln(P(v)) \tag{3.18}$$

In order to analyze long series, a windowed version is often used, which consists in computing the recurrence plots from overlapping windows of fix size. This is computationally much more efficient, while giving similar results in our case. As will be seen in the experiments section, different window sizes can actually be used to obtain good results. Our hypothesis about this result is that, in the case of scenes, relevant recurrences happen at a short time scale. These diagonal and horizontal lines will also be counted in larger recurrence plots.

In general audio recognition, the most useful feature aggregation is usually a vector composed of the global mean and variance of the frame-level features. In some cases, statistics of the derivative, which reflects the rate of change of the frame-level features, are also used. In this case, since we look at long series, the average rate of change does not help in recognition. However, since we look at short windows, it may be expected that other statistics can be computed, particularly the local variance of the features inside the window (the local means will obviously average to the global mean). As the experiments will show, both local variance and RQA features provide complementary descriptions of the short-term evolution of the spectrum, which can be used to increase accuracy with respect to using only global statistics.

Sound events

Segmentation As we have mentioned earlier, the distinction between events and background in sound scenes can be supported from a psychoacoustic perspective (Bregman, 1994). We view the physical world as a composition of objects that vibrate and produce sound as a result of different interactions. When large quantities of events overlap, we are no longer able to identify them, and we perceive them as background noise. From a musical perspective, sound events found in recordings are particularly

useful, as they can be used in a similar way as samples from traditional musical instrument notes. For example, any event resulting from an impact can be used as a percussive sound to make rhythms. However, while the effort required to record a scene is small, manually identifying and annotating different events is a tedious task. For this reason, blind segmentation of scenes to find events can be especially useful. While the identification of sound events has been investigated in several disciplines, from the perspective just described it has some different and specific requirements. For instance in MIR, many onset detection methods have been developed, either for analysis of polyphonic or monophonic signals. However, in MIR offsets are usually ignored, since the rate of onsets is typically fast (at least in the types of music that are commonly analyzed in MIR), and offsets are masked by reverberation and subsequent onsets. In speech recognition, Voice Activity Detection (VAD) (Ramirez et al., 2007) is usually performed by a binary probabilistic model that considers events and background. While this coincides with the perspective we have described, these models may not generalize beyond speech, as a weak sound can produce the probability to jump to a high level. This can be useful for a telephone conversation, where the background noise is suppressed but it is preferable to include it if there is any possibility of an informational event. However, in the case of fishing events for music in long recordings, it may be preferable to be more restrictive and select only the most salient events. Moreover, some VAD systems focus exclusively on voiced events by exploiting the pitched quality of the human voice (Tucker, 1992). Finally, most systems for environmental sound detection have been developed to recognize specific kinds of sounds in a supervised fashion (i.e., a machine learning model is trained for every kind of sound), which restricts their use for blind segmentation. Within the framework of cepstral coefficients, a simple but effective approach is to adapt the High Frequency Content (HFC) onset detection function that is often used in MIR. Among several versions, the simple weighted sum of the spectrum magnitudes proposed by Brossier (2006) performed best in our experiments. One of the main advantages of this function is that, since it follows the energy envelope it has a smoother decay and can be used to detect offsets better than other common functions. Our modification consists simply in approximating it from the output of the frequency domain gammatone filterbank described above. This reduces the computational cost, and actually results in better performance. Thus, from the matrix G of raw

filterbank features, we define the fliterbank-based HFC measure as:

$$HFC_{fb}(j) = \sum_i iG_{i,j} \tag{3.19}$$

Following Bello et al. (2005), we post-process the onset detection function in the following way: first, the function is z-scored, then smoothed with a 5 point moving average filter, and then a long term (400 point) moving median filter is subtracted from the resulting signal. Figure 3.6 shows this measure computed on a short indoor sound recording, (which is used for evaluation in section 3.4.4) compared to energy (obtained from the magnitude spectrum and post-processed in the same way) and a classic VAD implementation (Sohn et al., 1999)(in this case, post-processing is not performed since the function is a probabilistic measure). While the VAD algorithm performs best in our evaluation with this simple case, this algorithm is problematic for more complex cases, as shown in Figure 3.7 (a short clip of a recording in a bus). Clearly, the VAD function will tend to jump also for very low energy events.

From the HFC_{fb}, events can be isolated simply by cutting above some threshold (close to 0). This contrasts with standard practice in MIR, where onsets are detected as peaks in the onset detection function.

Aggregation Identified segments corresponding to events can be extracted from the sequence of frame level descriptors of the recording and described as separate entities. Like in the case of scenes, obtaining a fixed-length vector representation is especially useful for indexing sounds in large databases, as generic machine learning algorithms can be applied. In the case of sound events, aggregation can be regarded as a simpler task. If we think about the case of pitched musical instrument timbres, it is traditional to describe their sound in terms of the spectrum corresponding to the stable part. Thus, the average of the spectrum (or its compression in the cepstral coefficients) can be a starting point for describing the sound coming out of a resonant body. Classic studies on timbre (Grey, 1977) proposed the embedding of musical instrument timbre into a space composed of the log attack time (i.e. the logarithm of the time it takes for the signal to reach its maximum), the spectral centroid (the baricentre of the spectrum for one frame) and the spectral flux (the difference between two consecutive spectra). A generic description for sound events can be obtained by simple statistics of the cepstral coefficients with the addition of some measures of the energy envelope, which can be obtained from the first cepstral coefficient. Clearly,

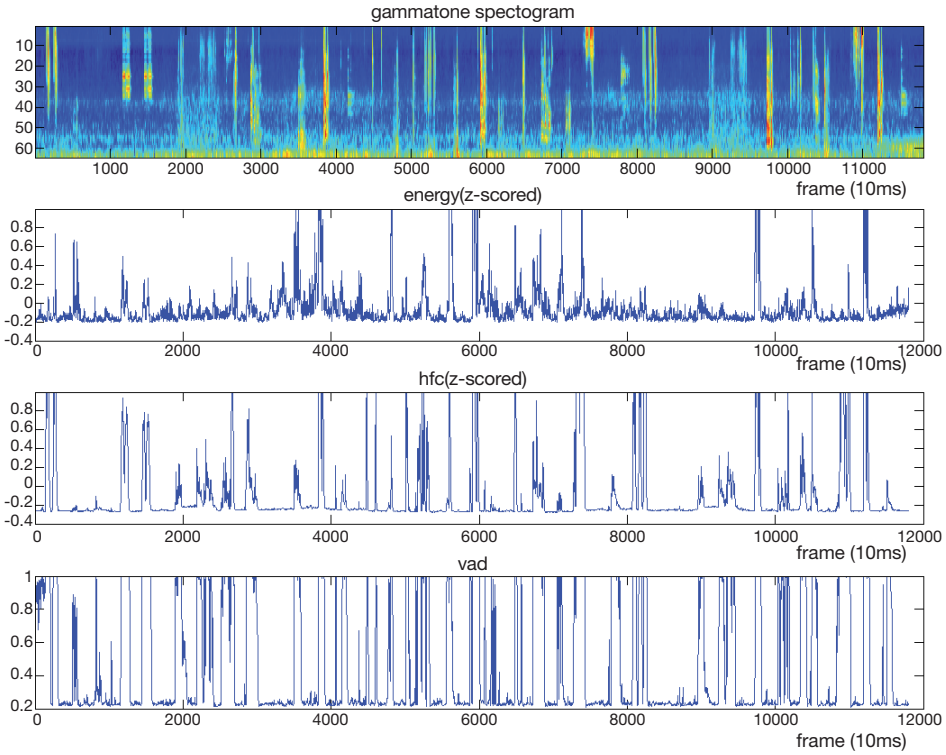


Figure 3.6: Example of onset detection functions

as in the case of the HFC measure, the spectral centroid could be approximated from the filterbank, and its cepstral coefficients can be seen as a more detailed description of the spectrum shape, where the centroid reduces it to one dimension. In a similar way, the statistics of the first derivative of the cepstral coefficients, which is often used in many recognition tasks, will provide more detailed information on the rate of change than the spectral flux. Hence, in addition to common statistics of the cepstral coefficients, we extract more detailed measures of the energy envelope, representd by the 0th GFCC: log attack time, temporal centroid, strong decay and temporal kurtosis and skewness(Herrera et al., 2002; Haro, 2008). Finally, RQA features can also be used for describing the temporal evolution of audio in short events. These features are more robust than envelope measures to different segmentation situations, which can be very varied in unstructured data.

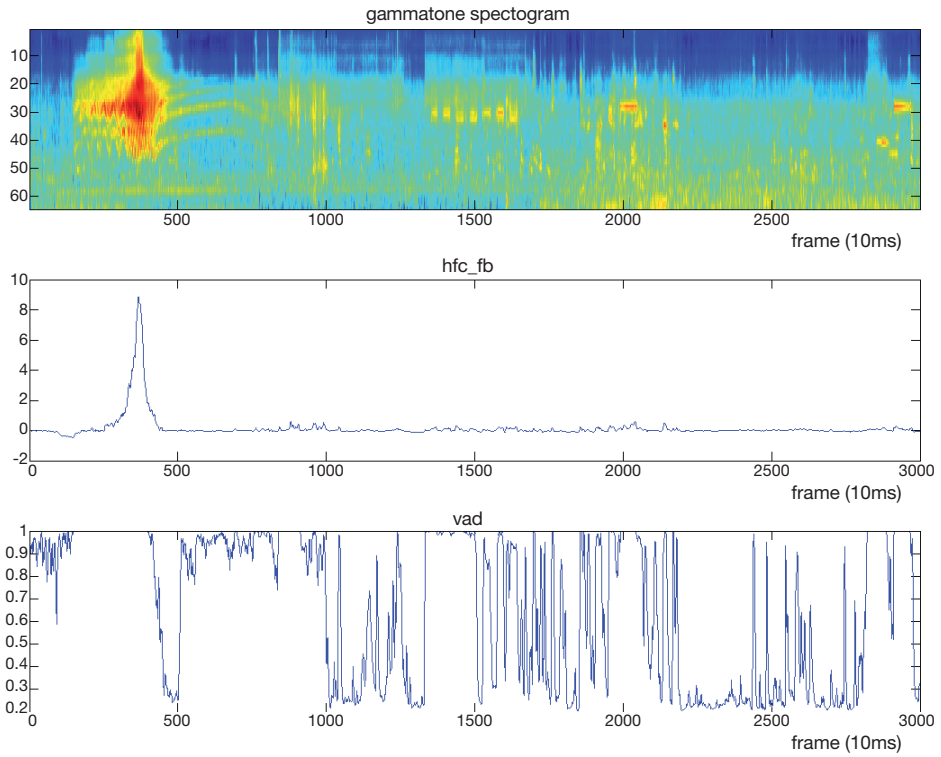


Figure 3.7: Example of the different behaviour of the VAD and the HFC_{fb} functions

Pitched events For music creation applications, the presence of pitch in sound events is obviously a very relevant aspect for many people. In this thesis, we focus on timbre, as the most general property of events, and the feature space that allows us to distinguish different sound sources (including musical instruments). Timbre is commonly represented using cepstral coefficients. However, by integrating the spectrum into filter bands of larger bandwidth, we loose the detail on the finer oscillations of the spectrum that correspond to the fundamental frequency and its harmonics. This may be regarded as a compromise: in order to describe timbre we must discard pitch and focus on the general spectral shape. Thus, for describing the pitch of events, we should go back to the spectrum. In our experience with the Freesound database, the spectral-domain version of the YIN algorithm, YIN-FFT (Brossier, 2006) can be useful for dealing with pitched events. This algorithm provides a measure of confidence that can be used as *pitched-*

ness, which is essential when dealing with unstructured data. Since pitch is already a musical feature, sounds with a stable pitch (such as musical notes) can be directly queried by their pitch value. In order to determine if a sound has a stable pitch, the variance of the evolution of pitch (computed in logarithmic pitch space) can be used along with pitch confidence from YIN-FFT. Apart from sounds with stable pitch, many sounds can exhibit characteristic pitch movements, such as *vibrato*, or *glissandi*. A systematic categorization of this kind of sounds is outside the scope of this thesis. The interested reader is referred to existing literature on morphological description of sound (Ricard, 2004; Peeters and Deruty, 2010). Finally, we may consider acoustic events containing multiple harmonic pitches, also known as chords. Indexing polyphonic tonality is also relevant in the case of music loops, as many loops shared by internet users have some defined tonality. Description of polyphonic tonality has been developed in the form of chroma features and Harmonic Pitch Class Profile (HPCP) features (Gómez, 2006). However these are generally applied in MIR, where tonality is assumed. In our case, a measure of tonal strength is crucial. A measure of key strength is the correlation of a HPCP vector with a tonality profile. However, this is not necessarily a good measure of tonal strength, since HPCP vectors of environmental sounds may correlate by chance. In our experience, indicators of peakiness of a probability mass function, such as Entropy, or Crest (commonly used to describe the peakiness of the raw spectrum) can be applied to HPCP vectors to identify chords, as these will display strong peaks. Since all pitched sounds will have some peak in the HPCP vector, the number of peaks can be used to distinguish chords from single notes. In general, features derived from HPCP vectors work well for unstructured audio where one may find all sorts of sounds, including monophonic and polyphonic melodies, notes, chords and chord sequences. However, a formal evaluation for this kind of sounds is out of the scope of this thesis.

Music loops

Loops are samples that can be played repeatedly to create rhythm. If we leave aesthetic considerations aside, anything could be a loop, as any audio sample will produce some sort of pulse when repeated. In practice, though, loops are popular in many electronic music styles, and for this reason it is common to find samples created with synthesizers and drum machines (and also traditional musical instruments) in online collaborative databases. Loops can be seen as monophonic or polyphonic segments that the author considers appropriate for repeating in a musical composition.

Compared to environmental sounds, loops are more complex and usually are based on musical features. Hence, many MIR techniques developed for beat and tempo tracking can be used to analyze loops. However, because of the traditional focus of MIR on music, there are several problems that complicate the use of many of these techniques: some require longer durations than what can be assumed for reusable fragments. On the other hand, many MIR techniques for rhythm analysis are based on onset detection so they rely on percussive or pitched events.

Foote's Beat Spectrum (Foote and Uchihashi, 2001) is a rather "classic" method for rhythm analysis. However, it has many interesting qualities that justify its use in the context of unstructured audio. First, like the rest of features that we have analyzed, it can be computed from the matrix of cepstral coefficients G_c (it could also be computed from the matrix of raw filterbank features G, but since it is based on distance computations it makes more sense to use cepstral coefficients). Thus, it allows us to detect and describe loops from the same base generic feature that we use for scenes and events. Second, since it is obtained from the similarity matrix of the sequence of cepstral coefficients vector, it does not make assumptions about the musicality of the sound, and it actually can be used to detect patterns in other feature domains such as HPCP. Similarly to the case of RQA features, we first compute the similarity matrix using cosine distance:

$$D_{i,j} = \frac{G_{ci} \cdot G_{cj}}{||G_{ci}||||G_{cj}||} \tag{3.20}$$

From this matrix, the beat spectrum can be obtained by summing all the diagonals in the matrix:

$$B(l) = \sum_{k=0}^{M-1} D(k, k+l) \tag{3.21}$$

where M is the length of the G_c time series and l is the time lag in frames. Recall that diagonals represent pairs of time points in the sequence of descriptors that share the same time lag, so a peak in the beat spectrum represents a typical repetition period in the underlying feature (in this case, timbre).

Figure 3.8 shows an example similarity matrix and the corresponding beat spectrum. From this base representation we can accomplish two important

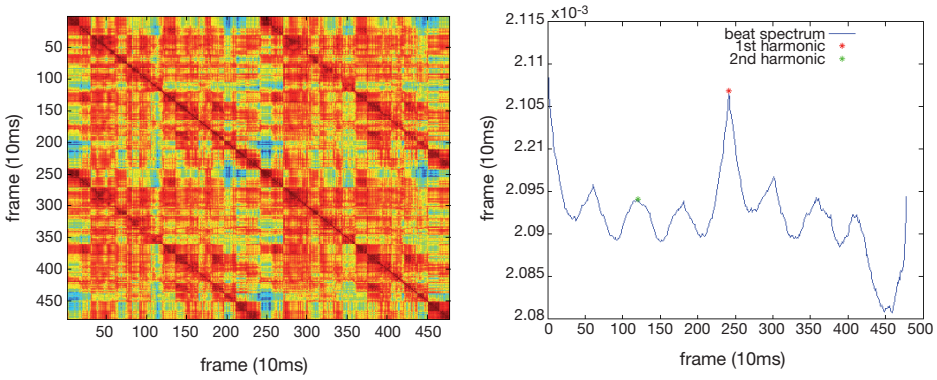


Figure 3.8: Similarity matrix and beat spectrum from a rhythmic sample

tasks in dealing with loops: we can identify them from other non-loopable samples in unstructured databases, and we can index them according to their most common repetition periods for rhythmic music creation applications.

Identification

Since the decision about what is and what is not a loop can easily become complicated, we propose a pragmatic heuristic to identify loops in unstructured data: we assume that loops will have some rhythmic content, and that they have been devised so that the duration of the sound has a harmonic relation with the main pulse of this rhythm. This is in practice what defines loops in electronic music creation since it will create the rhythm sensation when repeating the sound. Thus, we analyze the main N peaks in the beat spectrum and look for a peak whose position is harmonic with the total duration of the file (this is, the position of the peak is related to the duration by an integer ratio), with some error threshold. Note that in the case that there are silences at the beginning or end of the file, the file may not loop properly unless it is clipped, and it will be discarded by this method. These files can be preprocessed in order to remove silence. If we find such a peak among the most prominent ones, we can decide that the sound is a loop. Both parameters (the number of peaks and the threshold) can then be used to adjust the sensitivity and so that we are more or less strict in the selection of loops. As will be shown in the experiments section, slightly better detection can be achieved by using a SVM classifier with some fea-

tures extracted from the beat spectrum: in addition to the presence of a harmonic peak, the value of that peak (the spectrum is normalized by the peak at zero lag), and the entropy of the beat spectrum (Lartillot et al., 2008): since loops contain rhythmic pulses, the spectrum will be peaky, while environmental sounds with no clear pulse will tend to be flat. These measures retain the advantage of not being directly related with timbre or other musical features that could be used to distinguish between "typical" loop sounds (e.g. with timbres associated to drum or synthetic sounds) and environmental sounds, so that classification is purely based in rhythmic aspects of the sound, and any loopable sound (at least with regularities in the GFCC sequence) can be identified, regardless of its source.

Timbre classification

While defining discrete classes of loops may be a tricky issue, timbre classification is a common way to index them, often based on the description of the instruments used to produce the loop. Here, we use general statistics from the matrix of cepstral coefficients, like in the case of general scenes, with the difference that derivatives do provide useful information for classifying loops, while RQA features do not (from the set of RQA features described, we could be interested in the recurrence rate from a rhythmic perspective, but the beat spectrum already uses the diagonals in the similarity matrix with more detail than the quantized version of the recurrence plot). Results for timbe classification of loops are shown in the experiments section.

Tempo indexing

One of the most important aspects of music loops is the possibility to sync them to a given music context. Tempo in BPM is the most used measure. Here, we explore an alternative approach based on the beat spectrum. The idea is that since this analysis does not focus on percussive or specific instrumental sounds, it can be used to detect unexpectedly loopable material in recordings not originally intended for music creation. However, detecting the exact tempo may be tricky unless the beat spectrum reveals a very clear structure. Given the method used for detecting loops based on harmonic peaks, our approach simply focuses on finding loops that can be played together in sync. The intuition is that if two sounds are repetitive with the same repetition period, they will produce some coherent rhythm sensation when played in sync. An experimental prototype based on this idea is described in section 3.4.4.

3.4 Experiments

3.4.1 Overview

In this chapter we have described our framework for content-based indexing of community-contributed sounds based on audio descriptors. First, we have described the cepstral coefficients framework as the most ubiquitous frame-level description generally used for content-based retrieval of all kinds of audio. Then, for each kind of sound (sound scenes, sound events and music loops) we have described aggregation strategies, extracting the features that are most relevant for applications dealing with each kind of sounds.

This section presents several experiments to evaluate and compare the different possibilities and parameter sets in the described approaches. First, we describe several datasets used in our experiments. Then we compare the different types of bands, and their cepstral coefficients, to establish a generic base frame-level descriptor that can be used for all kinds of sounds. The advantage of this is obvious, as it allows analyzing large scale databases without knowing the types of sounds that they contain. In the following sections, we evaluate our approaches for identifying and summarizing sound scenes, sound events, and music loops.

Both supervised unsupervised indexing can be evaluated with datasets that represent different classes of sounds. Obtaining ground truth data is difficult, since consistently labelling recordings of different sounds with sufficient scale requires effort. However, it is essential to validate methods with as many datasets as possible in order to avoid overfitting specific characteristics of one dataset. We now describe several datasets obtained from different sources that are later used to evaluate the different proposed descriptors and algorithms. The purpose is to reflect the different use cases and types of data mentioned, so mainly they contain field recordings, sound events and music loops. Table 3.1 provides a summary.

d_case_scenes

This dataset was distributed to participants in the *D-CASE* challenge for scene classification (Giannoulis et al., 2013). It contains 100 30s. recordings corresponding to different environments: busy street, quiet street, supermarket/store, restaurant, office, park, bus, tube/metro, tube station and open market. The recordings were done using a binaural micophone during 2 months in the same geographical area.

in_house_scenes

This dataset was created by one of the authors of the publication where we first demonstrated the use of RQA features for sound scene classification in the context of the *D-CASE* challenge (Roma et al., 2013). The classes are the same as for the previous dataset, but recordings were obtained from different commercial CDs and online sources, and sound quality and bit rate varies between files.

dares_scenes

This dataset was collected by a team at the university of Groningen (Netherlands), using the same equipment for all recordings (van Grootel et al., 2009). The concepts are very similar to the *d_case_scenes* dataset, but the number of examples for each class is very variable. We selected the classes that had a minimum of 9 examples each.

d_case_events

The *D-CASE* challenge included an event detection task. The dataset supplied to participants for development of their own detection/classification algorithms consisted of two parts: a labelled database of events for training, and a set of three scenes with annotated events of the same classes of the training set. As explained in the previous sections, our system considers segmentation and classification separately, so the training dataset can also be used to evaluate classification of sound events. We then use the test files to evaluate the joint segmentation/classification framework.

gaver_events

This dataset was compiled from several sample CDs and sounds from Freesound by a team of 4 people (Roma et al., 2010). Sounds are classified according to the sound event taxonomy proposed by William Gaver (Gaver, 1993).

looperman

This dataset was crawled from the popular loop sharing site looperman.com for research purposes. The original dataset containing more than 20.000 sounds was sampled to a more manageable size so that different descriptors and parameter sets could be compared. All loops in the site are labelled according to “category”, “genre”, key and tempo. Categories describe the

main or single instrument in the loop, so we use these categories to evaluate timbre-based classification of loops. Tempo is used to evaluate the system described for rhythm indexing.

indaba

This dataset was downloaded from the music mixing site indaba-music.com. This site originally included an online, flash-based, DAW-style interface, which allowed basic multi-track audio editing, and promoted community activity mainly through remix contests. Recently the activity seems to be focused on the contests and challenges, and the DAW interface has been removed. The site includes a library of sound clips, although the descriptions are very sparse. The dataset contains only a few loops that are described according to instrument classes, like in the case of the looperman dataset, but much smaller in size. The loops also have a tempo annotation that we use to validate our rhythm indexing approach.

freesound tags

Sounds in Freesound are generally required to have a textual description and a set of at least three tags. Over the years, this has resulted on a rich folksonomy, in which some tags have become very popular. We sampled the 6 most popular tags: *field-recording*, *drum*, *voice*, *noise*, *loop* and *ambient*. The number was chosen as to avoid clear overlaps, as the next most popular tag is *percussion*. By definition, this dataset contains scenes, events and loops, and so it is only useful for the first experiments analyzing generic audio features.

freesound packs

In addition to the description and tags, sounds in freesound.org can be assigned to *packs*, which authors can use to group related files. These packs can then be downloaded as zip files. It can be expected that packs contain consistent similar sounds, not only because the author has followed some criterion, but also because the recording equipment or procedure will usually be the same or very similar. This dataset contains sounds from the 15 largest packs in Freesound. Like in the case of tags, it contains mixed sounds so it is only used to evaluate generic features.

Table 3.1: Datasets used for evaluation

Name	Classes	Instances
d_case_scenes	10	100
in_house_scenes	10	150
dares_scenes	8	72
d_case_events	16	320
gaver_events	9	1608
looperman	20	2000
indaba	3	371
freesound_tags	6	1034
freesound_packs	15	2164

3.4.2 Generic features

In this chapter we have described MFCCs as the most widely used descriptor for audio analysis. We have also seen that within the same framework there are several variants that have been used in other works. In this experiment, we compare mel bands, bark bands and erb bands for obtaining a generic representation. We also analyze the performance of raw filterbank features and cepstral coefficients for classification tasks.

Methodology

We compare the different feature sets using an SVM classifier for all the described datasets. In order to find optimal parameters for the base features, we first evaluate by averaging the raw filter bank features over time (i.e. the "bag of features") and analyze the performance with different numbers of bands, ranging from 10 to 128, for each type of band and for all datasets. Then we fix the number of bands and test in the same way for different numbers of cepstral coefficients. We then choose the number of bands and coefficients for the feature aggregation experiments.

In all classification experiments, we evaluate using a 10-fold cross-validation. For each fold, we run a grid search to optimize the γ and C parameters of the RBF kernel, training each parameter set on 20% of the training data. In order to account for the variability in the results, each classification task is run 10 times. We report the mean and standard deviation of the classification accuracy of the 10 runs.

Results and discussion

Figures 3.9 and 3.10 show the classification results for each dataset with the raw filter bank features. Shaded areas indicate standard deviation of the 10 runs. The first observation that can be made is about the different classification performances achieved with each dataset, with the most basic aggregation method - the average of the frame-level features. This indicates how easy is to discriminate each type of sound (given the number of training examples) on the basis of the average spectral shape. Thus, we can see that the *indaba* dataset (where classes correspond to musical instruments - hence timbre) can be easily classified. Also, the classifier achieves good performance with the *freesound_packs* dataset, which is expected to be a consistent division. Contrastingly, it is very difficult to discriminate among the main freesound tags. This is due to the mentioned problems with free tags. With respect to the type of band, results indicate that the difference between the three types of band is not significant in the smaller datasets (i.e. less than 1000 sounds). However, in the larger datasets, the erb-gammatone bands tend to perform better. Finally, it is easy to notice a saturation effect at about 40 bands, the standard number of bands used in the most common MFCC implementations.

We then repeat the test using cepstral coefficients computed from 40 bands. Figures 3.11 and 3.12 show classification accuracy for both groups of datasets. In comparison with the raw bands, by using cepstral coefficients results improve very noticeably. This is to be expected from their orthogonality, which results in more accurate distances. With respect to the type of band, again the slight improvement of erb bands can be observed only for larger datasets, with the exception of the more difficult *freesound_tags* dataset.

Finally, and perhaps surprisingly, there seem to be little or no significant gain in adding cepstral coefficients, in many cases from the very start. A safe choice seems to be around 25 coefficients. After this there is generally little improvement.

From these experiments we conclude that a generic descriptor of 25 coefficients computed from 40 ERB bands is a reasonable choice for general classification within the general framework of cepstral coefficients.

We now describe our experiments with respect to feature aggregation for each of the three types of sounds described: sound scenes, sound events, and music loops.

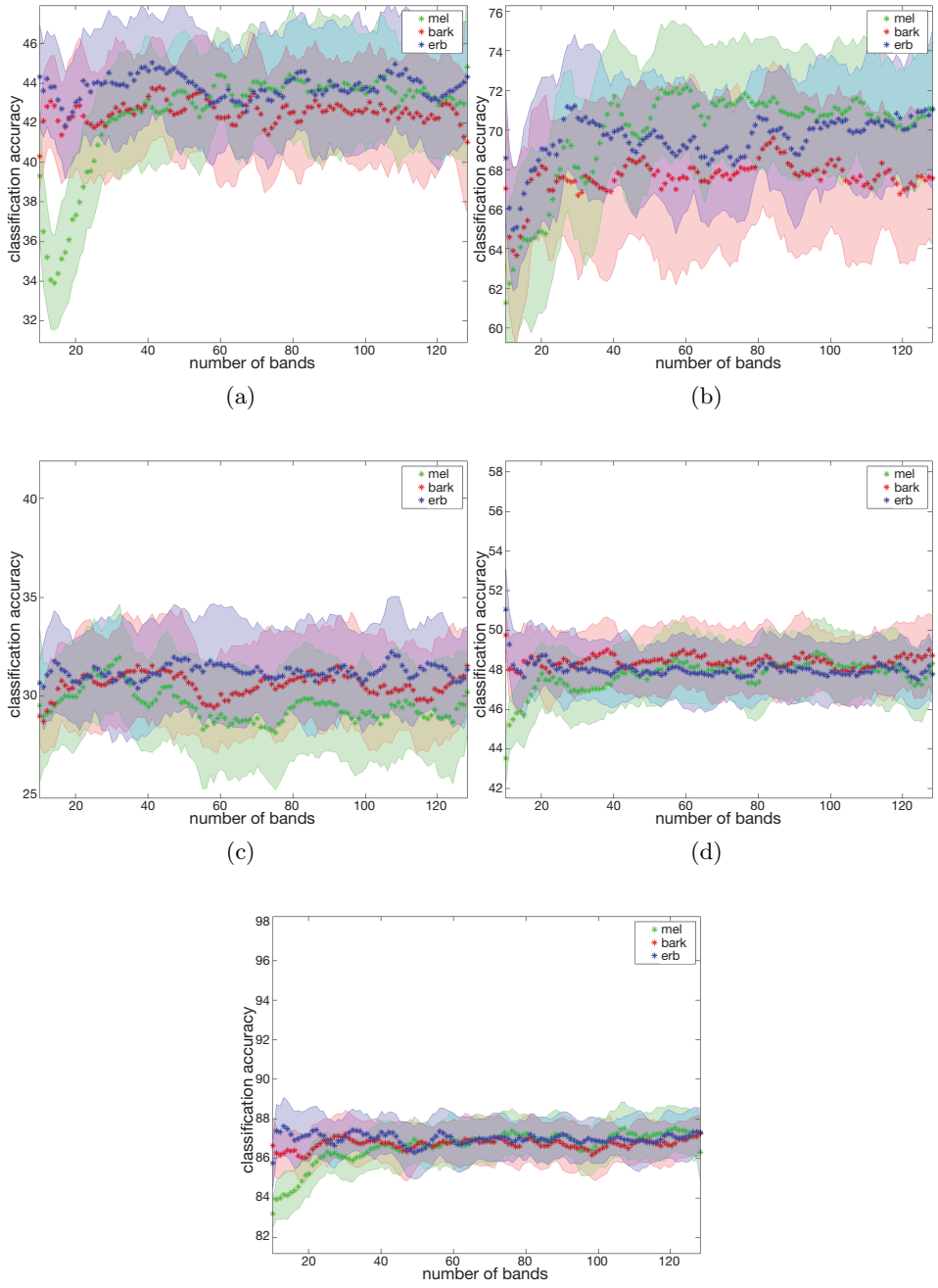


Figure 3.9: Classification accuracy using raw filterbank features for the smaller datasets: *d_case_scenes* (a), *dares_scenes* (b), *inhouse scenes* (c), *d_case events* (d) and *indaba* (e), as a function of the number of filters

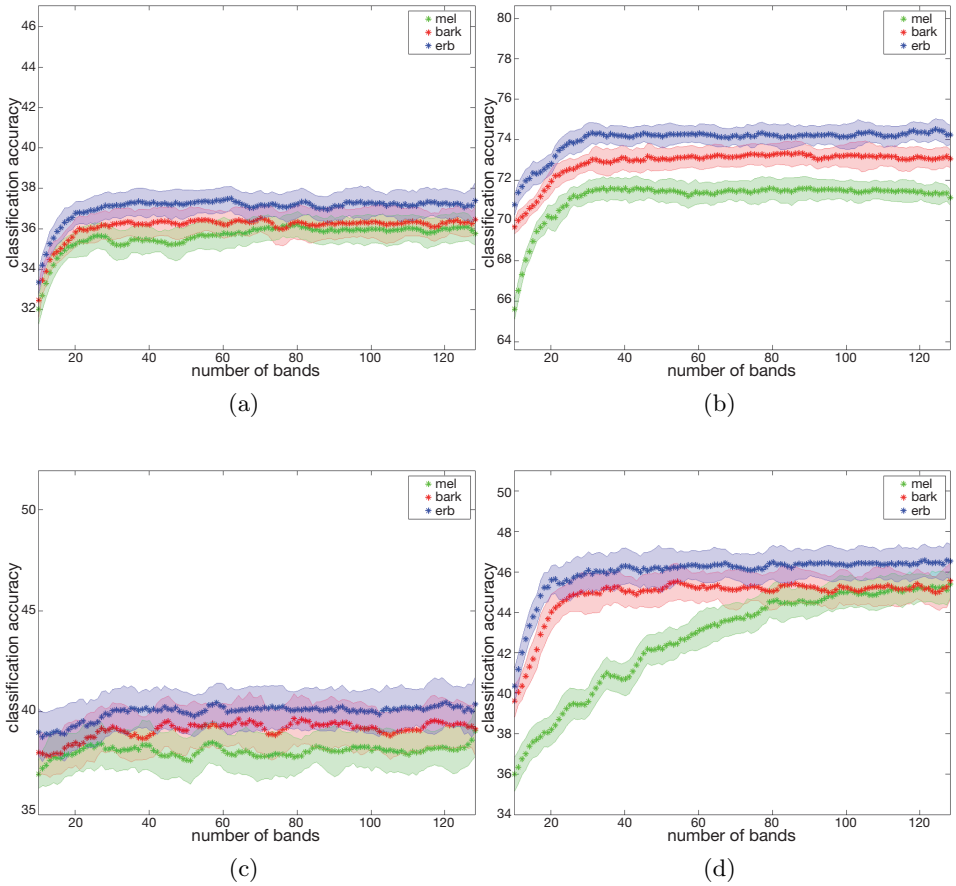


Figure 3.10: Classification accuracy using raw filterbank features for the larger datasets: *looperman* (a),*freesound packs* (b), *freesound tags* (c) and *gaver events* (d) datasets, as a function of the number of filters

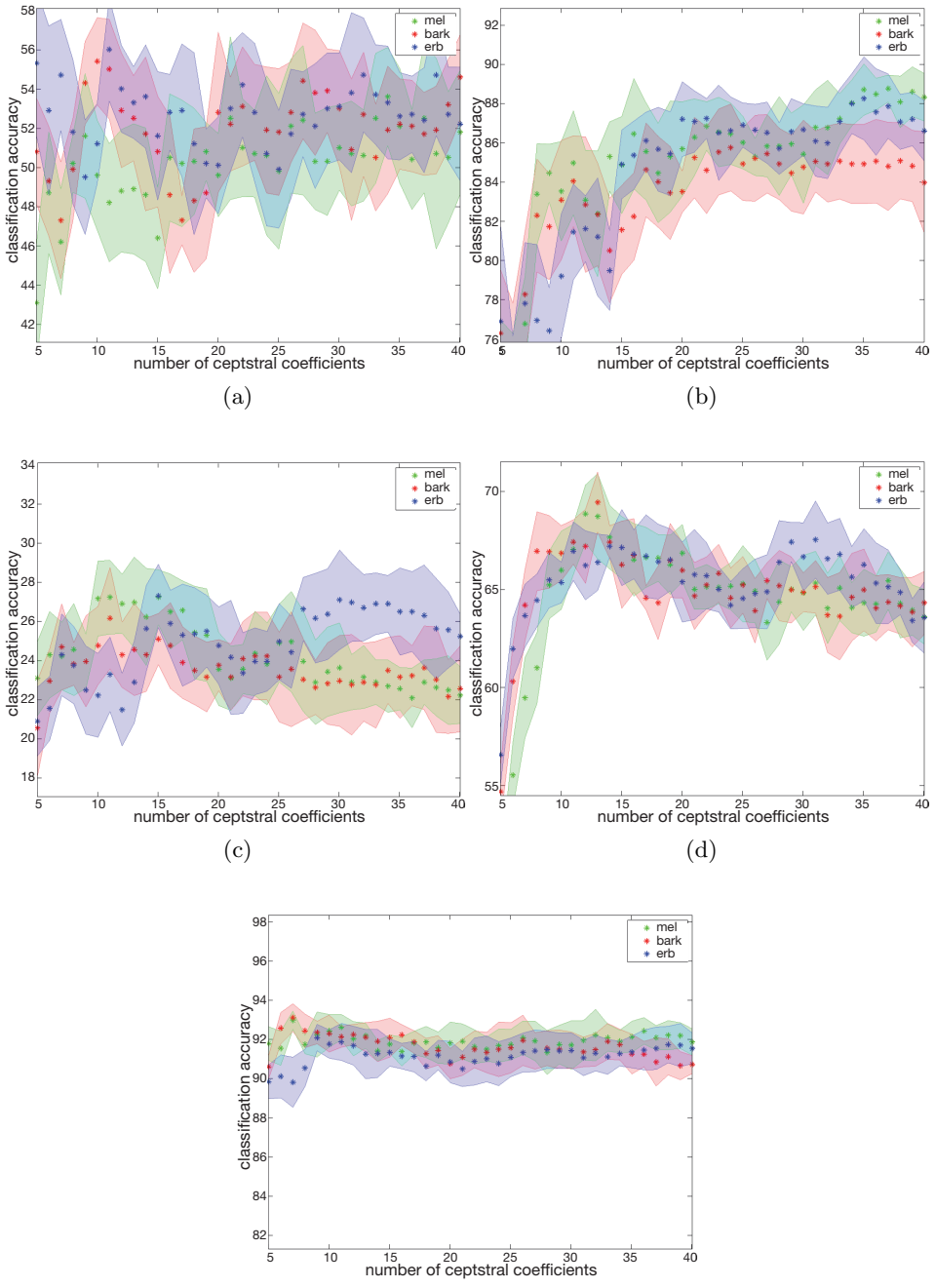


Figure 3.11: Classification accuracy using cepstral coefficients computed from 40 bands for the smaller datasets: *d_case_scenes* (a), *dares_scenes* (b), *inhouse scenes* (c), *d_case events* (d) and *indaba* (e), as a function of the number of filters

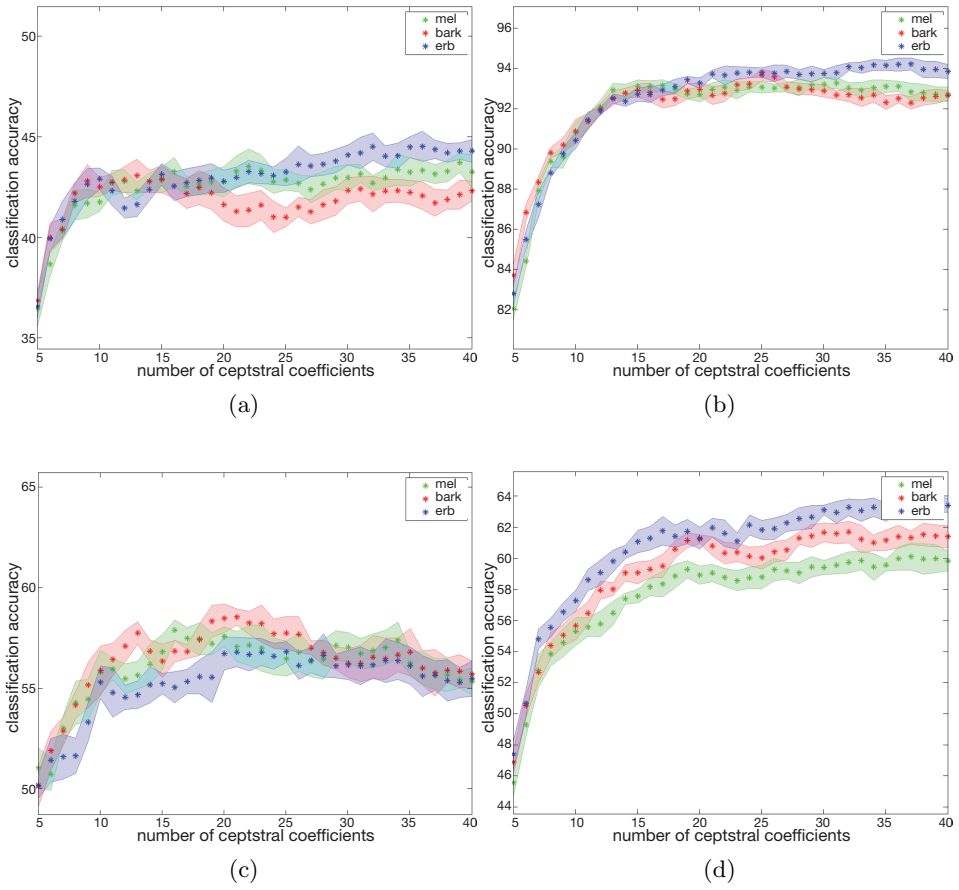


Figure 3.12: Classification accuracy using cepstral coefficients computed from 40 bands for the larger datasets: *looperman* (a),*freesound packs* (b), *freesound tags* (c) and *gaver events* (d) datasets, as a function of the number of coeficients

.

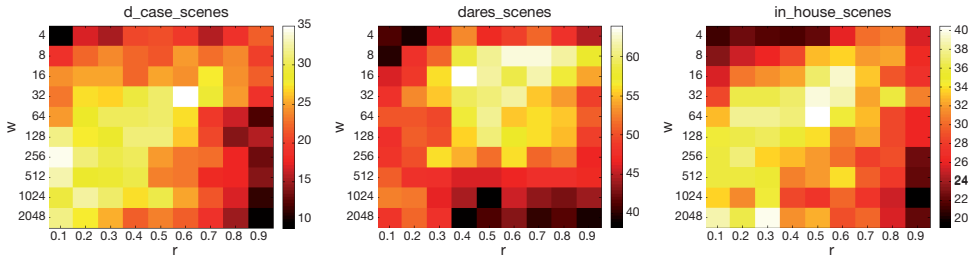


Figure 3.13: Classification accuracy using RQA features without whitening, as a function of parameters W and r for the scenes datasets

3.4.3 Sound Scenes

As we have described in previous sections, our approach to describing sound scenes is based on both their long term and short term statistics. To the best of our knowledge, the RQA features described for analysis of audio spectra have not been previously used for audio recognition (RQA features have been used with raw audio waveforms for pathological analysis of the voice (de A Costa et al., 2012), which is a very different case). For this reason, an analysis of the main parameters used for RQA may be relevant. The most important ones are the radius r (the distance threshold used to transform the distance matrix in the binary recurrence plot) and W, the size of the window within which the features are computed. Webber and Zbilut (1994) provide some hints for the choice of suitable parameters.

Methodology

We analyze the accuracy of a classification task using different sets of features and parameters for the datasets containing sound scenes (*d_case_scenes*, *in_house_scenes* and *dares_scenes*). The classification approach is the same as in the previous experiment. In order to find appropriate values for the parameters, we first perform a grid search for window sizes from 4 to 2048 spectral frames (we generally use 10ms hops between frames, so this means 40ms to 20s) and radius values between 0 and 1. We then compare the different sets of features for summarizing sound scenes.

Results and discussion

In the implementation of these features, we noted that z-scoring the features of each window to zero mean and unit standard deviation could produce

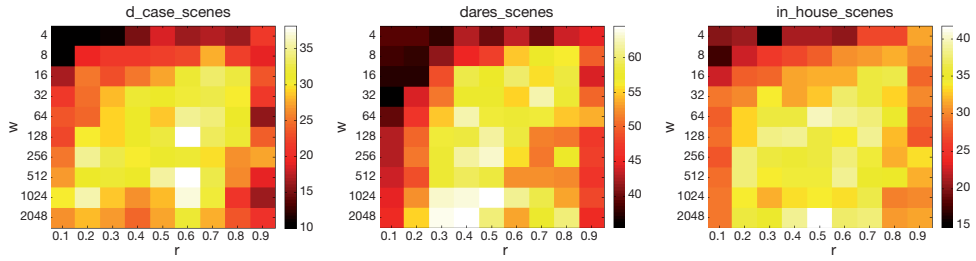


Figure 3.14: Classification accuracy using RQA features with whitening, as a function of parameters W and r for the scenes datasets

some small improvements. In the original RQA theory, the multivariate feature used for the distance matrix is obtained from an embedding of a one-dimensional feature, and this step is not used. The distance matrix is normalized by the maximum distance value, which we apply after computing the matrix from standardized features. One interesting aspect of this step, in our case, is that it makes it easier to choose a a fix set of parameters for all datasets. Figure 3.13 shows a color map of classification acuracy for a grid search of W and r using the three scenes datasets, where color represents the average classification accuracy over 24 runs. In this case only the standard normalization was used. Figure 3.14 represents the same search, using the feature whitening step. In the second case, the maximum classification accuracies are slightly better, and the areas formed by good parameter combinations tend to be larger. Also, the pattern seems to be more uniform across datasets. In general, by z-scoring features it is possible to select larger window sizes, as the resulting distances are not so dependent on the size of the window, and longer term recurrences can be counted. In the non-whitened case, the optimal window sizes are very small (in the order of 160 milliseconds).

The values for W and r can generally set to 128 frames and 0.6 respectively, for sound scenes. We now analyze classification accuracy using different feature aggregation sets: global mean (gm), global variance (gv), local variance (lv), RQA (rqa), and several combinations (gm + gv, gm+ gv + lv, gm+ gv + rqa, gm+ gv + lv + rqa). Figure 3.15 shows the results for each set of features. It can be seen that, with the exception of the *dares_scenes* dataset, local statistics can be used to improve accuracy when added to global statistics, and that lv and rqa increast the overall accuracy when added together, which indicates that they provide complementary information. In the case

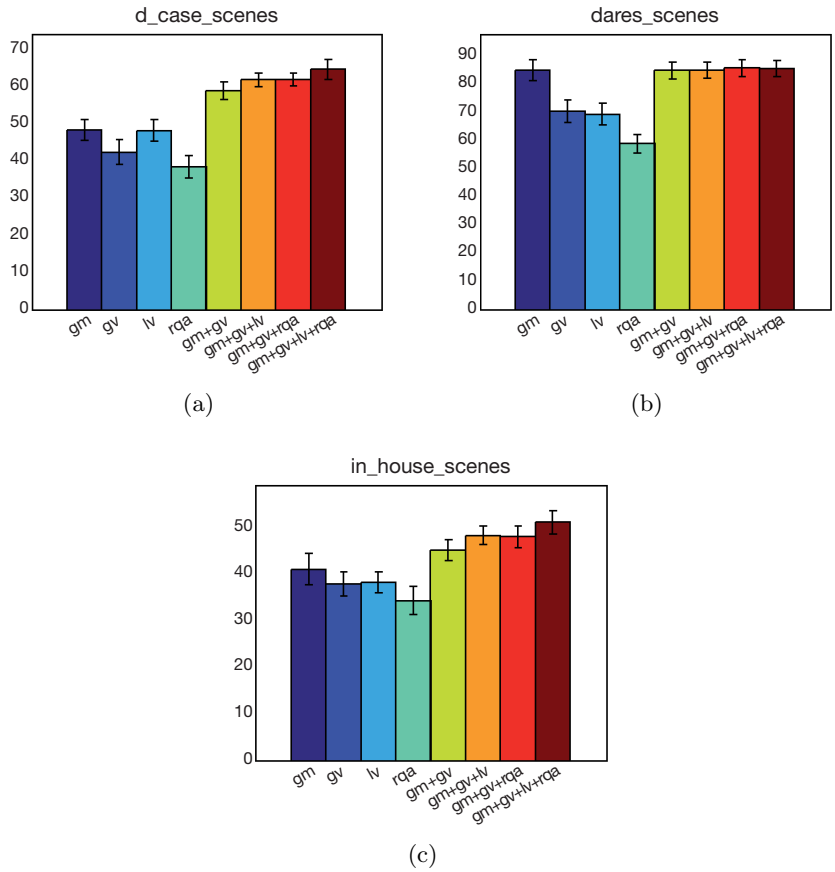


Figure 3.15: Classification accuracy using global mean (gm), global variance (gv), local variance (lv) RQA (rqa), and combinations for the scenes datasets

of *dares_scenes*, relatively good results are already obtained only by averaging the frame-level features, and no significant improvement is made by any other feature set, which seems to indicate a ceiling effect. Unlike the other scenes datasets, this dataset contains several classes of indoor scenes, which can be seen as a very different problem to recognizing outdoor recordings, as the background is usually not so relevant when identifying indoor scenes. The overall set of features, computed over 25 cepstral coefficients, amounts to a vector of 86 features for describing sound scenes.

3.4.4 Sound Events

We have described a method for identifying sound events in longer recordings, as well as different aggregation features based on cepstral coefficients. In the following set of experiments, we analye the tasks of blind identification and classification, and the recognition task combining both.

Methodology

The methodology follows the taks proposed in the *D-CASE* challenge (Giannoulis et al., 2013). We first focus on segmentation, comparing different algorithms for VAD and onset detection. We then focus on the classification of discrete events into classes, analyzing the performance of different feature sets. Finally, we perform both steps within the evaluation framework of the challenge.

Segmentation

For evaluating segmentation algorithms, we used the scripts provided in the *office live* task of the *D-CASE* challenge. These are 1 minute recordings of an office environment that contain several non-overlapping events. While the audio is not synthetic, it is not a completely realistic case, since it has been devised to avoid overlapping events and to ensure that all events correspond to classes in the ground truth. We convert the ground truth annotation, containing the locations and classes of the events, into a binary signal of the same length of the waveform, where 1 signals the presence of an event and 0 background. We then approximate the same signal using the different algorithms, and compute the jaccard distance between both signals. This distance measures the overlap between binary variables. Hence, the lowest the distance, the better the algorithm is approximating the ground truth. Table 3.2 shows the results for several common onset detection functions.

Clearly, HFC tends to perform better than the other onset detection functions because of the longer decay. Also it can be noted that the filterbank version performs even better than using the raw spectrum. Finally, the VAD algorithm generally gives good results in two of the scripts. It seems that for this task, using VAD would be the best option. However, this may be probably due to the simplicity of the evaluation data, since the scripts contain very sparse and recognizable events.

Table 3.2: Jaccard distance of different onset detection functions with the ground truth segmentation

Algorithm	script 1	script 2	script 3
HFC	0.2725	0.3478	0.2754
HFC_{fb}	0.2207	0.3030	0.2766
energy	0.4048	0.4900	0.5953
spectral flux	0.4394	0.5228	0.6252
vad	0.2224	0.2553	0.3194

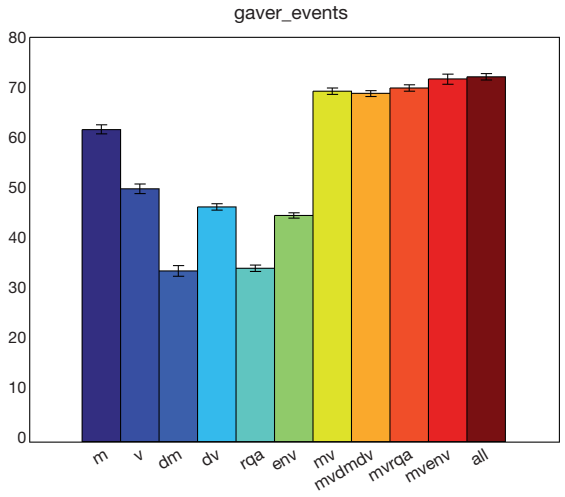
Classification

We now analyze several features for classification of events. We use the training set provided for the *D-CASE* challenge, and the *gaver_events* dataset. The following descriptor sets are compared: mean (m), variance (v), mean derivative (dm), variance of the derivative (dv), envelope descriptors (env), RQA (rqa), and several combinations of these (mv+dm+dv, mv+rqa, mv+env, all).

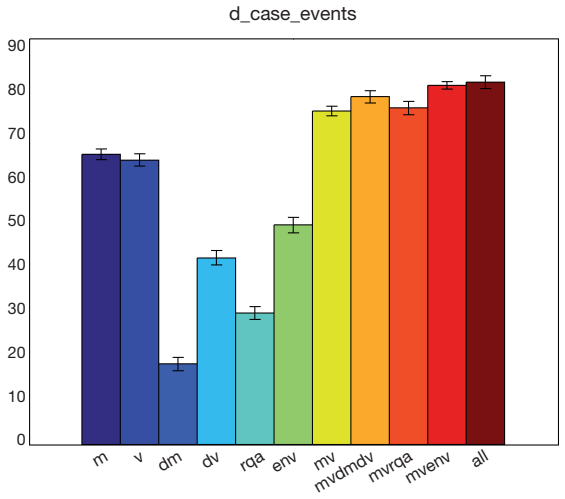
Figure 3.16 shows classification accuracy for each set of features with the two events datasets. It can be clearly seen for both databases that the envelope features provide relevant information, achieving greater accuracy than the derivatives and the RQA features. The derivatives only seem to work for the *d_case_events* dataset, but not for the larger *gaver_events* dataset. Aggregating all these statistics results in accuracies above 70%, far above the corresponding random baseline in both cases.

Recognition

We now combine both tasks into a *recognition* task. In this task, we segment the same scripts as in the previous section, and classify the resulting



(a)



(b)

Figure 3.16: Classification accuracy using mean (m), variance (v), derivative mean(dm), derivative variance (dv), envelope features (env), RQA (rqa), and combinations for the events datasets

events into the different classes defined in the training set. In this case, we adopt the methodology used in the *D-CASE* challenge, and analyze the frame-based F-measure of the classifier. This means that using the script recordings used in the segmentation experiment, we consider whether each 10ms frame has been correctly assigned to its labelled class (note that many different evaluation measures were used in the challenge(Giannoulis et al., 2013), we selected this measure for simplicity). Figure 3.17 shows the results for the different feature sets and segmentation algorithms used in previous experiment, including also "ideal" segmentation (i.e. the ground truth annotations are used for deciding the event boundaries instead of a segmentation algorithm). In general, results seem to differ slightly from the classification experiment, and not so much with respect to the segmentation experiment. It should be noted that in this case we are analyzing a relatively small test set (in the order of 30 events per script, results being averaged across three scripts) where the classes are not balanced. Contrastingly, the *gaver* events dataset used in the classification experiments contains more than 1500 instances. Interestingly, RQA features provide an improvement in most cases, which was not so clear in the plain classification task, while the event-based features tend to perform worse. Since this also happens in the ideal segmentation case, this difference seems to be more related to the presence of background noise in the script recordings than due to segmentation issues. With respect to the segmentation experiments, differences are not very important. Considering all three experiments, and our qualitative insights on the performance of the VAD algorithm, it seems that the HFC_{fb} segmentation with RQA features (in addition to traditional statistics) can be a reasonable choice for event detection. However, the generality of these results are limited by the size of this dataset.

Music loops

With respect to loops, we have identified three different tasks required for indexing loops in the context of unstructured audio databases. The first one is identifying loopable samples. Remind that in this case we focus on loops created and shared (but not necessarily with an appropriate description) by internet users, so our problem is identifying loop files from other non-loopable files. We then analyze classification of loops based on timbre, in the same way we cosidered events and scenes. Finally, we analyze the beat spectrum method for tempo-based indexing.

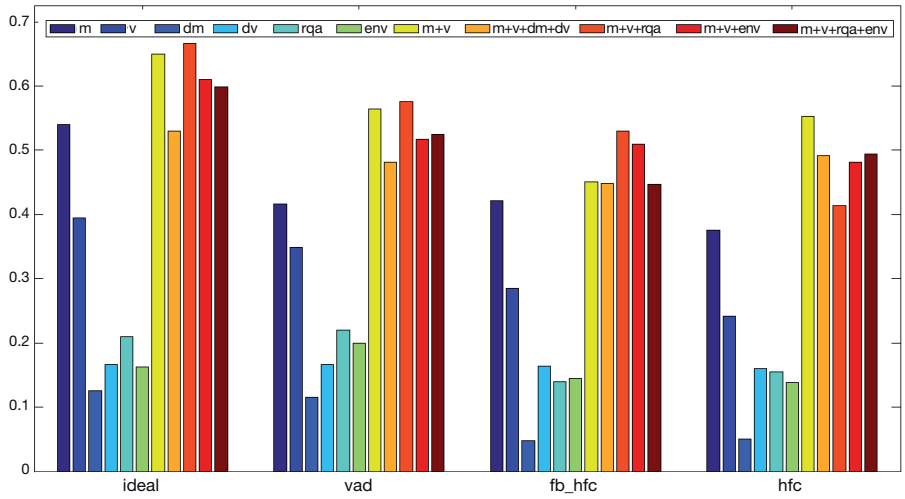


Figure 3.17: Frame level F-measure for event detection using different segmentation algorithms and feature sets

Methodology We evaluate the proposed algorithms in a set of experiments. While the classification task is based in the same methodology of the previous experiments, we followed different approaches for the case of identification and rhythm indexing. We describe the methodology in the corresponding sections.

Identification of music loops In order to evaluate our method for identification of loops, we created two *ad hoc* datasets divided in 100 loops and 100 non-loopable files. The first dataset was obtained by querying the *freesound* database for sounds tagged as *loop*, and then using a general query and avoiding sounds that have this tag for the negative class. In addition, the "group by pack" of Freesound's search engine is used to avoid sounds from the same pack, and in each case, 100 sounds are randomly sampled from a larger result set to eliminate any effects of the order of search results. The second dataset consists of 100 sounds from the *looperman* dataset and 100 from the *in_house_scenes* dataset. We compare two different strategies: in the first case, classification is purely based on the first harmonic heuristic: if one of the 20 top peaks in the beat spectrum is a harmonic of the file duration (with an error threshold of 0.1), then the sound considered a loop. In the second case we add the beat spectrum entropy and the value

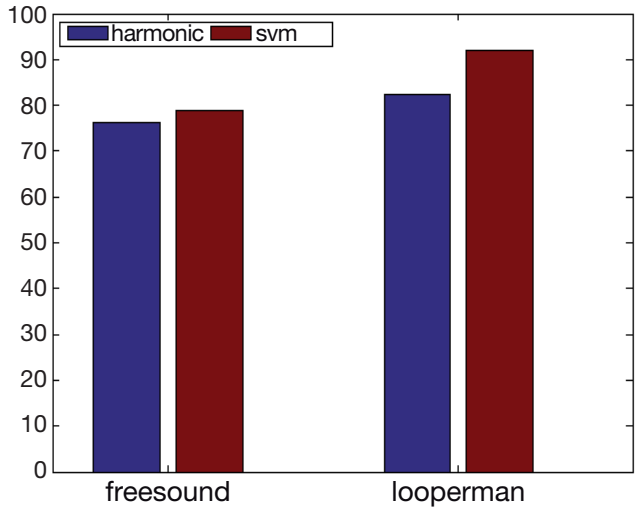


Figure 3.18: Loop detection accuracy using either the harmonic method and the SVM classifier for the looperman and indaba datasets

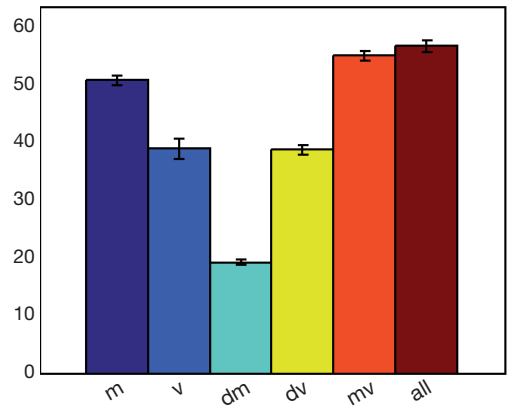
of the harmonic peak (along with the former binary variable) as features to a SVM classifier, and evaluate via 10-fold cross-validation. Results are shown in figure 3.18. It can be seen that this simple approach gives very good results in the case of the *looperman* vs *in_house* scenes dataset, with a significant improvement when using the SVM classifier. In the case of the freesound dataset it should be noted that labels are noisier: a sound not labelled as "loop" may be actually loopable even if it's not described as such, and sounds labelled as loops may be incorrectly cut. This method should give good results in practice, as false positives (sounds that were not intended to be loops) will generally be loopable material, and false negatives (discarded loops) will possibly lack rhythmic clarity, at least in the timbre domain. Further improvements can be done by analyzing the beat spectrum of other features, such as HPCP.

Timbre classification While defining discrete classs of loops may be a tricky issue, timbre classification (i.e. often based on the instruments or sounds they contain) is perhaps the most common way. Both the *looperman* and the *indaba* datasets contain non-overlapping labels related to musical instruments. We analyze the potential for classification in the same way we did for the scenes and events datasets. Results are shown in figure

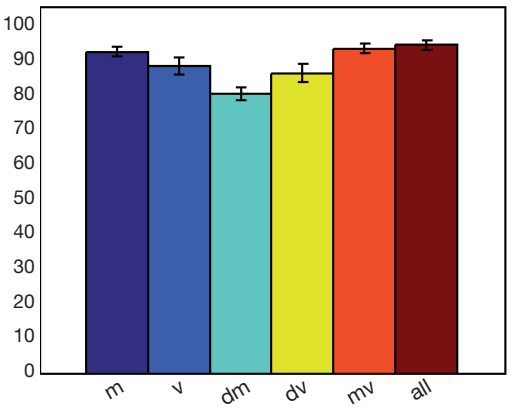
3.19. Clearly (as can also be understood by listening to some examples), the labels in the *looperman* dataset are not as consistent with the actual sounds as it may seem. This suggests that the problems inherent in free tagging are not necessarily solved by imposing a set of fixed categories in the site interface. In the case of the *indaba* dataset, both the number of classes and the number of instances are much smaller. Unlike in the case of scenes and events, statistics from the derivative of the cepstral coefficients seem to add relevant information in both cases, which is consistent with their extended use in MIR. Since loops contain mixtures of different instruments or sounds, unsupervised classification could provide more interesting results than supervised classification based on single-instrument labels. Another common approach that could be automated would be electronic music genres, which would require a large and consistent training dataset and joint analysis of timbre and rhythmic features.

Rhythm indexing As we have described, the beat spectrum does not directly give BPM values, but the harmonic peaks used to identify loops will very likely be physically related to the main rhythmic pulse perceived in the loop. Here, instead of focusing on BPM values, we can just index loops according to their characteristic repetition periods. The following experiment can help showing the potential of this method. The main assumption is that, as long as loops can be played together, in a creative context octave errors do not matter much. This is, a loop labelled at 60 BPM can be perfectly played with another one labelled as 120 BPM. Since we search for harmonics of the file duration (which can contain one or several bars), the first harmonic (typically one half or one third of the file duration) will be a period containing several beats. We analyzed 200 sounds from the looperman dataset and computed the ratio between the duration of this period and the duration of one beat according to the labelled tempo in BPM. We then computed the error as the decimal part of this ratio. Figure 3.20 shows the distribution of the error for 500 sounds, with most sounds below 0.05 (mean 0.09, sd 0.12). This indicates that the periods are generally related to the labelled tempo. Typically the ratio between the beat spectrum harmonic and the duration of one beat derived from the BPM is 2, 4 or 8.

Experimental prototype We implemented a web-based prototype in order to test the proposed method for rhythm indexing, using loops from Freesound. While we didn't conduct a formal evaluation, the development of the prototype helped clarifying the potential and limitations of the pro-



(a)



(b)

Figure 3.19: Classification accuracy using mean (m), variance (v), derivative mean(dm), derivative variance (dv) and combinations for the loop datasets: looperman (a) and indaba (b)

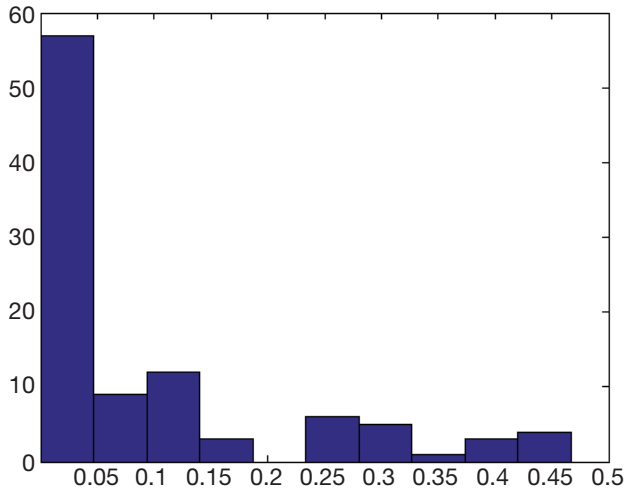


Figure 3.20: Distribution of the error in detecting labelled tempo harmonics with the looperman dataset

posed approach. The index, computed as described above, contains about 30.000 loops. We used several information visualization techniques in order to present loops in a web environment. First, we computed a histogram of all the harmonic periods of loops in the database. Since the beat spectrum is computed from frame-level descriptors, harmonic periods are measured in frames, and the histogram is discrete. Peaks in this histogram correspond to frequent periods related with typical tempo values (e.g. 120BPM). This is a consequence of the social nature of the database. The histogram is used as an interaction device, so that selecting one bar from the histogram would load all sounds that share the same harmonic period. When queried for sounds with a given period, the server creates a k-nearest neighbors graph of the sounds according to their timbre similarity, as computed by cosine distance over the average of MFCC features. The graph is returned to the client, where a force-directed layout is used to display all the sounds in the page. Since the graph is built according to timbre similarity, the layout will tend to group similar sounds together, which helps in the exploration process. It should be noted that this is an approximation to our clustering approach described in chapter 4, as modulartiy has been shown to be related to force-directed layout (Noack, 2009). This kind of layout has been used in the context of corpus-based synthesis for large databases (Schwarz

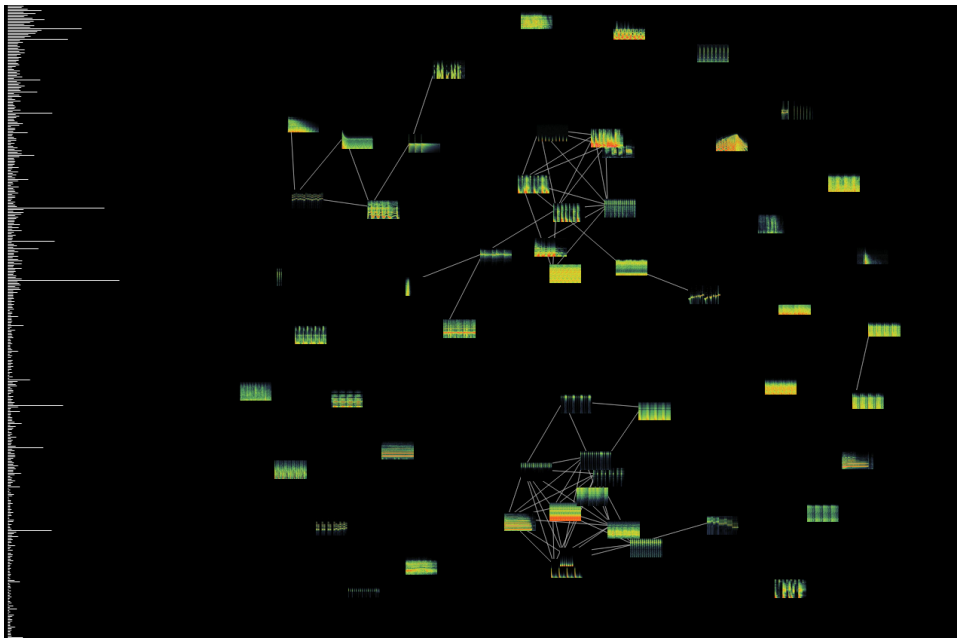


Figure 3.21: Screenshot of the prototype

et al., 2009). Loops are represented via the small spectrogram thumbnails available for all sounds in Freesound. This allows the user to quickly identify sounds with a strong pulse, as they will contain vertical lines. Sounds with melodies are also easily identified. Playback of the loops is quantized to the selected loop period.

While the prototype is still in development, reactions in initial demonstrations were very positive. The interface allows maintaining a certain creative flow, as all sounds and can be used as a musical instrument, as well as an exploration of the sounds uploaded by the Freesound community. The main limitation is that some sounds are indexed according to harmonic periods that do not necessarily start at the beginning of the file. Foote proposed that the "phase" of the beat spectrum can be computed as a novelty curve from the similarity matrix (Foote, 2000), which can be seen as an onset detection function. This feature could be used to improve the index.

3.5 Conclusions

In this chapter we have faced the problem of low level audio representation in the context of user-driven databases without a pre-defined editorial process. In such platforms it is possible to find any kind of sound, including speech, music, environmental audio, and all possible mixes of these. Among the literature of the respective disciplines that study each kind of sounds, perhaps the most ubiquitous low-level representation are MFCCs. Other common descriptions, based on more recent studies of the auditory filter, such as bark or ERB bands, can be incorporated in the same framework to obtain cepstral representations. Our experiments show that swapping the filter bank does make a difference, although not very large, especially with larger datasets. This seems to imply that more accurate filterbanks motivated by psychoacoustic studies allow for better generalization. We have then proposed to distinguish three main kinds of sound files that are common in online audio databases: sound scenes (usually field recordings), sound events, and music loops. Within the framework of cepstral coefficients, we have proposed a way to identify each kind of sound, and for each kind, we have analyzed different ways to aggregate the time series of cepstral coefficients in order to obtain a summary representation. These representations can be used for indexing and retrieval for creative applications, such as described in Chapter 4. At the same time, we hope that these results can be useful in different fields, particularly in analysis of environmental sound recordings, where it is also possible to find all kinds of sounds.

Chapter 4

Automatic taxonomical organization of audio

4.1 Introduction

During the last decades, many web applications have allowed users to share information generated or captured by themselves. These applications are typically less structured than traditional media with well defined editorial processes, curated by companies with well defined role hierarchies. This feature can be related to the success of the decentralized design of the Internet and its protocols. The notion of unstructured data is commonly associated with text. It can be argued that web search engines have helped people to become accustomed to interacting with unstructured and diverse information by way of short text queries. Most popular social media applications, such as *flickr* or *youtube*, are rather laid-back with respect to information organization. Free tags have become a popular way to organize content, even for private use. From the content creation perspective, an emphasis on structure may be problematic: most web users who create and share content usually do it for fun and are not likely to commit themselves to the tedious work of labeling their data in a very detailed or repetitive way. Moreover, decisions about the global information structure may be questioned and subject to debate. Wikipedia can be seen as an intermediate model, which works according to an elaborate governance system. It is quite obvious that the possibility of indexing and retrieving unstructured data has boosted Internet adoption and thus helped disrupting the way we use information. At the same time, a bit of agreement and coordination are

helpful for making sense of information on the web. Very often, the use and mis-use of unstructured data services call for something more elaborate. As an example, Freesound.org heavily relies on free tags. Describing sounds is usually a tedious task, and many users try to minimize their efforts by repeating the same tags and descriptions for many sounds. Some popular tags include 'A', '120' and 'Zoom H2', which refer respectively to a musical note or key, a musical tempo, and a portable audio recorder model. These and many similar examples suggest that a more structured model would help to describe the sounds. The Semantic Web vision (Berners-Lee et al., 2001) has received a great deal of attention in academic Information Retrieval. One core idea of the Semantic Web was that if information was described in compatible ways across different web services, we would be able to make programs that automatically relate information from different sources and process it in intelligent ways. For this purpose, labels should conform to web ontologies, which allow to formally describe relationships between concepts in an application domain. The adoption of Semantic Web technologies, however, has been slow. In a way, the irruption of users into content creation and the generalization of unstructured processes conflicts with the complex formalisms proposed by Semantic Web researchers. In this chapter we explore taxonomical organization as an intermediate solution that is well known to most users and can be seen as a simple form of ontology.

4.2 Taxonomical organization

Taxonomical organization is perhaps one of the oldest ways of structuring knowledge, and is central to most scientific disciplines. Any computer user is accustomed to organize files in a hierarchical fashion. Taxonomies may be unnecessary for some applications. For example, current mobile operating systems (which are increasingly used by people with little or no experience with personal computers) tend to deliberately hide the file system from the user, and applications such as gmail offer free tags as opposed to traditional folder structures. In this sense, taxonomies can be thought as a compromise between the complexity of semantic web ontologies and the randomness of free tags. Taxonomies are, of course, a simple form of ontology, but one that all computer users can easily understand, and that can be created by domain specialists without a computer science background. Taxonomical organization of sound has been studied from many different perspectives. Before electronic means of sound production existed, the sounds used in

music were limited to traditional instruments, and musical timbre was associated to instruments. Musical instruments were organized into taxonomies usually according to their production mechanism, which influences the kind of sounds that the instrument can produce (Kartomi, 1990). An example is the Hornbostel-Sachs taxonomy, which is widely accepted in Western countries. At the beginning of the 20th century, avant-garde artists started to see that technological progress would allow creating music with different sounds. In his manifesto for a Futurist Music, Russolo (Russolo, 1986) described a taxonomy of urban sounds that suited the futurist aesthetics, and invented a number of instruments (the *intonarumori*) that could produce those sounds. In his *Traité des Objets Musicaux*, Schaeffer (1966) discussed how sounds could be described and categorized, but from a more open perspective which included many different criteria. In the 1970s, Schafer (1977) also proposed a taxonomy of sounds in *The tuning of the world*. Like Russolo's, Schaffer's taxonomy is heavily influenced by aesthetics, and includes impossible or unheard sounds (such as the sound of creation or the sound of apocalypse). In the 1990s, while researching on sound icons, Gaver (1993) proposed a general taxonomy of sounds motivated by acoustic ecology. The general idea is that in everyday listening (as opposed to musical listening) we use sound to extract information from the environment, and so we try to understand the mechanism of sound production. Thus, he proposed a general division of sound according to interactions between basic materials (Figure 4.1). While in theory his idea is to cover all kinds of sounds, the taxonomy is especially suited for environmental sounds, as it does not deal with the specific significance of pitched sounds such as human and animal voices, musical instruments or alarms and signals. Although a universal taxonomy of sounds seems an impossible task, Gaver's proposed taxonomy is simple enough and has been widely adopted in sound design (Van Den Doel et al., 2001; Rocchesso and Fontana, 2003; Hermann et al., 2011). In this chapter, we adopt this taxonomy as a paradigmatic example of top-down approaches for labeling audio, which can be supported by supervised algorithms for content-based audio indexing. In general, taxonomies can be devised specifically depending on the application, including instrument taxonomies, electronic genres for loops, sound effects, and so on.

4.3 Content-based indexing

Text (or more specifically hypertext) is the original and still most prominent type of content of the web. Text in web pages is used by most search

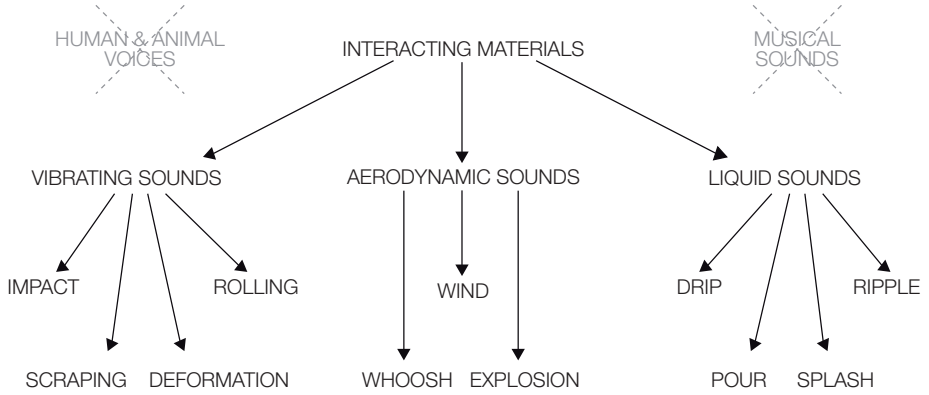


Figure 4.1: Gaver's proposed taxonomy of sounds

engines for indexing all kinds of content. However, there are several problems with using textual descriptions in unstructured, user-contributed sound databases, such as Freesound, for music creation. Textual descriptions, for instance in the form of social tags or keywords, are always incomplete and inconsistent. The same word is used by different people in different ways. Sounds obtained in radically different ways and with very different semantics may have very similar perceptual qualities, while sounds with the same description may be completely different. As an example, a query for "car" will retrieve the sounds of car horns, engines, doors and crashes. Perceptual qualities of sounds, rather than text, are usually considered as the main musical material. This was, after all, the first observation of Pierre Schaeffer when considering the use of recordings for music creation. Schaeffer proposed the concept of "reduced listening" in order to abstract and isolate the perceptual aspects of sound from the semantic references to the sound source (Schaeffer, 1966). For these reasons, in this thesis we focus on content-based approaches to audio retrieval. Content-based audio retrieval and discovery is based on automatic descriptions obtained from the audio signal. Such descriptions are intended to represent perceived qualities of the recorded sound. On top of signal descriptors, it is common to use machine learning techniques to label sounds with higher-level concepts. In the case of creative applications, simple and abstract labels can be used. An analogy can be drawn with traditional musical theories, where some concepts (such as e.g. notes in western music or bols in indian music) are used merely as indicators of some perceptual sound quality that has been parti-

tioned into discrete classes. The distinction between this kind of classes and more complex semantic categories can be related to the distinction between perceptual and categorical learning in psychology (Mandler, 2000). Our approach consists in grouping user-contributed sounds in discrete classes that can be assigned simple text labels for reference. From a machine learning perspective, these groupings can be obtained using two main types of methods: supervised and unsupervised.

Supervised approaches are implemented as top-down processes: they start from an already established set of labels and try to assign some of these labels to audio fragments. This requires the availability of a training set of already labeled sounds, from which the algorithm must learn how to label the rest. The amount and quality of training data has a critical impact on the result. However, obtaining large quantities of training data can be a complex and time-consuming task.

Unsupervised approaches can be seen as bottom-up processes, where the groupings are assumed to be present in the distribution of the data. Thus, they don't require additional training data. However, the quality of the result will also depend on the amount of data and their distribution (e.g. whether sounds are evenly spread with respect to a certain quality). It can also be challenging to interact with a set of sound clusters without referencing any specific theory or model.

4.4 Unsupervised indexing

Unsupervised indexing is useful for analyzing data that has not been labeled or classified. As we have mentioned, bottom-up, collaboratively-built databases can be seen as "unstructured data", which are not organized or curated according to a predefined structure. In these cases, even when labels such as free tags are used, noise appears as a consequence of different points of view, expertise levels, goals and attitudes. A data clustering algorithm, able to find inherent structures that are not explicit, can be especially useful in such situation. When derived from content-based descriptors, partitions generated by data clustering can be especially suited for music creation, as sounds are grouped according to their acoustic similarity, which seems a natural way to conceptualize and manipulate sound collections. Clustering algorithms can also be used to automatically index long recordings with different events and/or sections that have not been labeled, in combination with the event detection methods proposed in chapter 3.

Classic algorithms are usually divided between partitional (such as *K-means*) and hierarchical (e.g. agglomerative clustering) (Jain and Dubes, 1988). However, classic hierarchical clustering algorithms are impracticable for large datasets due to their computational cost. On the other hand, many algorithms are not well suited for noisy data, especially when the feature space has very different densities. As an example, in a collaborative audio database, it is quite likely that many users will upload a particular kind of sound (e.g. bass drum), while other types of sound are scarce. For such data it is common to use graph-based algorithms. Graphs based on the number of Nearest Neighbors (k Nearest Neighbors or kNN graphs) can adapt to areas of different densities, since no fixed distance is assumed. This approach also helps with the *Curse of dimensionality* related with the size of the feature space, as features are not directly used in the clustering step. In recent years, the concept of modularity has originated a number of graph partitioning algorithms, each with different characteristics, and for which open source implementations are available. These algorithms can be used to automatically index sounds according to different needs, including taxonomical organization.

4.4.1 Construction of the kNN graph

As mentioned, a common approach in document clustering is to construct a graph that links each document to its k nearest neighbors. Clusters can then be identified by partitioning the graph. The advantage of the kNN approach is that it allows identifying clusters of different densities. In an area with high density, neighbors will be chosen among very close documents, while in areas of low density, the nearest neighbors of a given point may not be so close.

One common problem for large datasets is that the construction of the graph may require computing the whole distance matrix, which becomes quickly prohibitive in both space and time complexity. One solution is to use some Approximate Nearest Neighbors (ANN) data structure, such as a Kd-tree (Bentley, 1975) or a Cover tree (Beygelzimer et al., 2006). Such strategies have also been applied to direct navigation of large audio databases for creative applications (Schwarz et al., 2009). In early experiments, we found that this approach provided similar results to computing the whole matrix with euclidean or cosine distance. Another problem is that for many dimensions, the distance measure may become meaningless. This may be partially alleviated by using the cosine or Jaccard distance in-

stead of euclidean. Another solution, used in the Shared Nearest Neighbors (SNN) algorithm (Ertoz et al., 2002) is using the number of neighbors as a distance measure. However, in our preliminary experiments this did not improve the results in the case of audio clustering.

There are several variants: the constructed graph may be directed or undirected, weighted or unweighted. For weighted graphs, the original distance or the number of shared neighbors can be used. In our experiments, ignoring weight tended to give best results. With respect of the directedness of the graph, the choice depends on the algorithm used for partitioning the graph. For example when using algorithms based on random walks, such as *walktrap* (Pons and Latapy, 2005), directed edges are important.

4.4.2 Clustering

Once built, the kNN graph can be treated as a complex network, and community detection algorithms can be used to find clusters. The most common methods are based on modularity optimization. Modularity evaluates a given partition of the network by counting the number of links between nodes in the same partition compared to their total degree. Here we consider modularity of an undirected multigraph. Given the adjacency matrix A of a graph where A_{ij} is the number of links between nodes i and j, modularity is defined as

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(v_i v_j) \tag{4.1}$$

where k_i is the degree of node i, m is the total number of edges in the network, and $\delta(v_i v_j)$ is a function that returns 1 if node i and node j are in the same group and 0 otherwise.

While direct optimization of modularity is computationally hard (Brandes et al., 2006), many approximate algorithms have been proposed. Some of them (such as *Walktrap*) can output a dendrogram and thus can be used like traditional hierarchical clustering algorithms. A very fast modularity optimization method that specifically searches for hierarchical community structures is the so-called *Louvain* method (Blondel et al., 2008). This algorithm works as follows: each node in the network is initially a community on its own. Then each iteration is divided in two phases. In the first one, each node is moved to its neighboring community that produces the highest modularity. In the second phase, communities are simplified into nodes. The iterations continue until no further gain in modularity is obtained. This produces a multi-level community structure that is not

restricted to a binary tree, and thus can be used as a taxonomical organization without further processing. While the complexity of the algorithm is not known, the authors assess that it "seems to run in $O(n \log n)$"[1]. The Waltrap algorithm depends on the sparsity of the graph, but it is expected to run in $O(n^2 \log n)$. In our experience, the *Louvain* method can be used to cluster the whole of the Freesound database (200.000 sounds) in a few seconds (in addition to computing the graph). Walktrap can be used for smaller subsets but it is limited by space complexity (expected to be $O(n^2)$). Other modularity optimization algorithms are not so well suited for these scales. In all, clustering audio by modularity optimization of kNN graphs is a very versatile method: it is almost parameter-free (except for the number of neighbors), it can be used to generate full dendrogram (from which an arbitrary number of clusters can be obtained), and it can also be used to find the most likely division of the data (i.e without specifying the number of clusters in advance) by using the division with maximum modularity.

4.5 Supervised indexing

4.5.1 Automatic classification

Supervised Machine Learning algorithms have seen great development in recent years. With the evolution of computer hardware and the possibilities for accumulating large quantities of information, such algorithms are becoming crucial to facilitate automatic organization of information. We focus on automatic classification algorithms, which aim to predict the class (or label) of some piece of information, in our case audio recordings. Supervised algorithms learn from a set of training examples that have been manually labeled, and are able to predict the labels for unlabeled signals. Labeling the training examples is a labour-intensive task, and the quality of a supervised system depends critically on the size of the training set for applications with high variability and dimensionality such as audio indexing.

Classification algorithms have been extensively applied to musical audio signals in MIR, for tasks where some sort of consensus is assumed. Examples include musical genres (Guaus, 2009) or mood (Laurier, 2011). A similar approach can be used when indexing audio recordings for the purpose of music creation. For example, supervised techniques have been applied to classification of musical instrument sound samples (Herrera et al., 2002; Martin and Kim, 1998). However, outside of musical instruments, it may be diffi-

[1] `http://perso.uclouvain.be/vincent.blondel/research/louvain.html`

cult to find a social agreement on how to classify sounds. As mentioned, this can be observed by analyzing the use of free tagging in communities such as Freesound. In general, the choice of a classification scheme can be regarded as an application-dependent decision. As an example, the Looperman community does rely on a fixed set of mutually-exclusive labels that allow classifying music loops into categories where one instrument dominates (i.e. bass loops, piano loops, and so on). For such cases, the use of supervised classification algorithms could significantly lower the cost of labeling sounds, as the categories could be automatically learnt and suggested by the system by training a classifier on an initial set of labels. While a number of machine learning classification algorithms exist, we adopt a generally accepted intuition in the MIR community: that most development is needed in the area of features, as opposed to new or more specialized classification algorithms (Herrera-Boyer et al., 2006). During the last decade Support Vector Machines (SVM) have become overwhelmingly popular, proving very effective in many domains, including text (Joachims, 1998), images (Chapelle et al., 1999), video (Schuldt et al., 2004) and audio (Guo and Li, 2003). In each area, researchers need to find an appropriate set of features that the SVM can use. In this sense, we have already evaluated different sets of features through SVM classification in chapter 3. A detailed description of the base SVM algorithm is outside the scope of this thesis, but a basic description can be useful to introduce the different options and parameters in our experiments.

SVM is a linear classification technique, which is extended by the so-called "kernel trick" to non-linear feature spaces. In SVM classification a training example is represented using a vector of features x_i and a label $y_i \in \{1, -1\}$. The algorithm tries to find the optimal separating hyperplane that predicts the labels from the training examples. A hyperplane with parameters w and b that separates two classes is defined as:

$$y_i(w^T x_i + b) \geq 1 \tag{4.2}$$

Since data is often not linearly separable, it is mapped to an infinite dimension space by a kernel function where such separation is possible. The most common choice is a Radial Basis Function (RBF) kernel with parameter γ:

$$K(x_i, x_j) = e^{(-\gamma|x_i - x_j|^2)}, \gamma > 0 \tag{4.3}$$

Using the kernel function, the *C-SVC* SVM algorithm finds the optimal hyperplane by solving the dual optimization problem:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{4.4}$$

subject to

$$\begin{aligned} 0 \leq \alpha_i \leq C, i = 1, ...N \\ y^T\alpha = 0 \end{aligned} \tag{4.5}$$

where Q is a $N \times N$ matrix defined as $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and e is the vector of all ones. C is a cost parameter that controls the penalty of misclassified instances given linearly non-separable data.

This binary classification problem can be extended to multi-class using either the *one vs. one* or the *one vs. all* approach. In the former approach a classifier is trained for each pair of classes, while in the later the classifier is trained for each class using examples from all the other classes as negative examples. The *one vs. one* method has been found to perform generally better for many problems (Hsu et al., 2001). In the experiments section, we use the *libsvm* (Chang and Lin, 2001) implementation, which uses *one vs. one* classification. Suitable values for C and γ are found through grid search with a portion of training examples for each experiment.

4.5.2 Hierarchical classification

With the extension of the classification problem to multi-class classification, there are two straightforward possibilities for obtaining hierarchical classifications. Perhaps the most intuitive approach is to train a classifier for each ramification. Taking Gaver's taxonomy as an example, this would imply having a model to discriminate sounds from solids, liquids and gases, and then three more models to distinguish the different types of interactions within each level. The main problem with this approach is that when predicting the class of a given unlabeled sound, errors produced at the first level will add up to errors in the second level. Hence, an alternative approach suitable for small taxonomies is to train a single model for the lowest level, and simply infer the location of the sound in the other levels from the definition of the taxonomy. In our experiments with Gaver's taxonomy, this approach tends to give better results.

4.6 Experiments

4.6.1 Human vs. computational factors

As we have discussed, in this chapter we explore the use of content-based techniques to organize large databases of sounds for music creation. Our focus is not necessarily obtaining semantic labels, in the sense of describing the source of the sound, but obtaining abstract categories that are consistent with respect to their acoustic properties.

We performed a preliminary user experiment in order to test this idea of abstract *audio lexicons* as content-based groupings of sounds generated through machine learning (in this case graph-based clustering). Our aim was to understand what are the factors that influence the ability of users to learn a particular lexicon. To this end, we analyzed the task of predicting to which grouping does a given sound belong. Particularly, we expected that qualitiy measures from the clustering solution would be determinant for this task. We were also interested in what human factors would influence the result.

Most of this work was published in (Roma et al., 2012b).

Methodology

Clustering The experiment was carried out before the development of the modularity-based approach described in section 4.4. We used *Chameleon* (Karypis et al., 1999), an algorithm based on kNN graphs that optimizes the *minimum cut* of the graph. Roughly speaking, a minimum cut is a partition of the graph that crosses the minimum possible number of edges. A partial implementation is available in the closed-source CLUTO package (Karypis, 2002). In contrast, open source implementations of several modularity optimization algorithms are available for different platforms. In order to obtain a test dataset, we first clustered a large database of 10.000 sounds obtained from Freesound, all shorter than one second in order to avoid sounds with many sound events and timbre variation.

Internal quality measures are often used to evaluate the obtained clustering solution. One common approach is to analyze the similarities within each cluster to understand how compact it is, and the similarities between points of each cluster and all points in the others to measure how well separated are the different clusters in the solution. For this purpose, the *vcluster* program in the CLUTO package outputs z-scored values of the similarity (in

our case cosine similarity) between each point and the rest of points in its cluster (internal similarity) as well as to the points in other clusters (external similarity). From these values we computed several quality measures: $C_{isim}(C_n)$, $C_{imax}(C_n)$ and $C_{imin}(C_n)$, respectively the mean, minimum and maximum similarity between points within cluster C_n. The mean similarity gives an indication of how compact is the cluster overall. The minimum similarity corresponds to the maximum distance between two points, which can indicate the presence of outliers. The maximum similarity in the cluster can be used as a hint of the maximum density inside the cluster, i.e. if the maximum similarity is not very high then the cluster will be sparse, whereas a dense cluster with some outliers will have a high maximum similarity even if the average is not so high.

Analogously, $C_{esim}(C_n)$, $C_{emin}(C_n)$ and $C_{emax}(C_n)$ are based on external similarities, i.e. similarities between points in C_n and points in all other clusters.

Determining an optimal number of clusters is a non-trivial issue that was not the focus in this work. Note that the number of clusters can be chosen automatically using the *Louvain* algorithm as described in section 4.4. In this case, we determined empirically a number of clusters that provided consistent results while allowing a manageable size for the lexicon (in the order of e.g. the size of the latin alphabet or the number of keys in a keyboard). In this preliminary analysis, it became clear that a larger number would give smaller and more consistent clusters. Yet, in real world applications we can not expect the user to learn hundreds of sound categories. We ran our algorithm to produce 40 clusters. Of these, we discarded clusters with less than 50 instances and chose a random sample of 6 clusters for each user. This number seems well aligned with acceptable cognitive load in short term memory (Miller, 1956). Of these clusters, we randomly chose 6 example sounds for each and again chose 20 random sounds from the pool of chosen clusters as test for that user.

Prototype For the experiment, we implemented a simple prototype on a Microsoft Surface multi-touch table, using Adobe Air technology (Figure 4.2). The interface showed 6 colored rectangles representing the different clusters, each of which could be unfolded to visualize and play the sounds. Hence, the only potential visual cues with respect to the clusters were the sound waveforms of the examples. These images are the same that are used for sound lists in Freesound (except that we removed the color). Test examples were showed as a pile of black and white waveform objects resembling

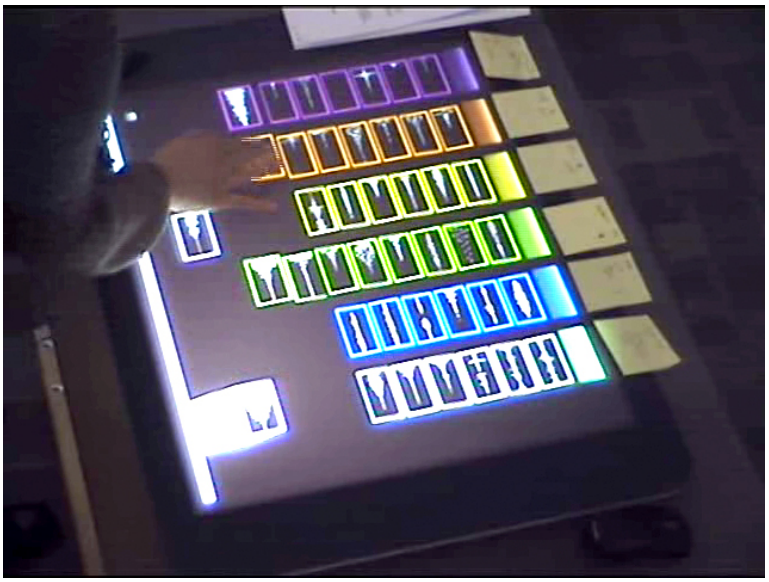


Figure 4.2: Picture of the prototype

a deck of cards. Dragging each card to the vertical area below the color shape corresponding to each cluster colored the object to the cluster color, which signaled that the sound was assigned to that cluster.

The experiment was divided in two main tasks. In task 1, participants were asked to listen to the sound examples of each cluster. In addition, they were asked to annotate any words or tags needed to identify and remember each cluster in sticky paper notes that were attached to the table above each cluster area. This allowed us to analyze how the users understood the clusters. In task 2, participants were asked to classify the stack of test sounds into the clusters according to their similarity by dragging them to the appropriate area.

The use of a large multi-touch table provided a more embodied interaction that allowed us to observe and analyze participants movements and strategies. Video was recorded with two cameras positioned non-intrusively: general view and close-up view. Video output from the device was also captured for complementing the analysis.

Finally, a questionnaire was filled in with basic demographic information, as well as some questions about their own confidence in the performed task and the criteria they followed for the classification.

The study took part in the Computing department of the Open University (UK). Most of the participants were familiar with computers, although not necessarily with music or audio-related topics. In total there were 14 participants (9 males, 5 females) with ages from 21 to 50 and a diversity of nationalities and cultural backgrounds. Most had some kind of musical training: 4 reported no training at all, 4 some degree of music training and 6 of them more than five years. With respect to familiarity with electronic music creation, 8 of them reported no previous experience, 5 of them had some familiarity, and one of them was a regular user of electronic music creation tools.

Data analysis After the experiment, we had several sources of data for analysis. The prototype logged the classification choices performed by users, and kept the information about the clusters that were presented. On the other hand, the footage from the cameras was used for video analysis, which is a common tool in human-computer interaction studies (Jordan and Henderson, 1995). This technique allows for a great deal of detail if compared to more traditional HCI methods, but requires time and a clear focus with respect to the problem at hand. We followed observations made during the experiment and an initial overview of the video to define our target variables (described in the next section). The *Elan*[2] software was used for the video analysis. The main advantage of this program is that it allows hierarchical specification of the code for annotating the video. We used the annotations, along with the data from the questionnaire, for qualitative and quantitative analysis of user-related factors. Finally, we used the data from the clustering and the logged results for quantitative analysis of factors related with the clusters.

Results and discussion

We analyzed both qualitative and quantitative aspects of the experiment in order to understand which factors could determine the correct assignment of sounds to their own cluster as computed by the clustering algorithm. This assignment was encoded as a binary variable for each of the tested sounds. Each participant was given 20 sounds, so in total there were 280 assignments. We aggregated the results in order to analyze the data from the point of view of the user and from the point of view of the clustering algorithm. In the first case, the target variable was the fraction of correctly

[2] http://www.lat-mpi.eu/tools/elan, developed at the Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands Sloetjes and Wittenburg (2008)

Table 4.1: Variables obtained from the experiment

Cluster level	
C_{size}	Cluster size ($\|C_n\|$)
C_{imean}	Average internal similarity
C_{imax}	Maximum internal similarity
C_{imin}	Minimum internal similarity
C_{emean}	Average external similarity
C_{emax}	Maximum external similarity
C_{emin}	Minimum external similarity
User level	
S_{t1s1}	Pre-listening of all table
S_{t1s2}	Listening to neighbor clusters
S_{t2s1}	Listening to examples again

assigned sounds by each user, and in the second, the fraction of correctly assigned sounds for each cluster. Table 4.1 shows a summary of all variables resulting from the experiment. To understand human factors related to the proposed task, we did a qualitative analysis of the responses to the questionnaire, as well as an analysis of the video footage. The fraction of successfully assigned sounds in each user oscillated around 40% ($mean = 0.44$, $sd = 0.5$).

We found several recurrent themes in the analysis of the questionnaire and paper notes, as well as the responses referring to the criteria used to classify the sounds. They are summarized in Table 4.2. Most popular criteria could be classified as *Sound sources* and *Sound properties*.

In the video analysis we observed some relevant aspects related with the behavior of participants. Our main observation was that participants followed different strategies in both tasks. In task 1, there were 3 participants who started by pre-listening to all or most of the sounds on the table before starting. This group devoted more time in the two tasks, and one of them scored the best result. Another difference was found between those participants who explored neighbor clusters before labeling a given one, and those who did not. In task 2, we observed that some participants tended to rely on their written notes, while others went back to listening to the examples when deciding to which cluster they would assign the sound. All but one

of the participants who correctly assigned more than 50% of the sounds followed this strategy of constantly comparing the test sound with examples. We confirmed the significance of these differences through two-sample t-tests over the three binary variables: users who followed strategy S_{t1s1} (pre-listening to the whole table) took longer ($p < 1e^{-15}$) and did better ($p < 1e^{-04}$), as well as users who followed strategy S_{t1s2} (looking at contiguous clusters) ($p < 1e^{-07}$). The result for S_{t2s2} showed that the group that kept listening to sounds performed better ($p < 0.02$). We further counted the number of times the example sounds and test sounds were played for the first 10 test sounds (after that, users tended to classify without playing the examples again). The overall count of plays of the examples in task 2 for each user correlates with the recognition rate ($r = 0.64$), while the correlation of number of plays of the test sound was very small ($r = 0.122$). In all, we were able to extract more significant variables from the learning phase (task 1), and the relevant outcomes in the classification phase still seemed to refer to learning strategies, which reflects the importance of this step.

We focused with some more detail on the three participants that scored best. All of them referenced concepts related to sound sources as their criteria in the questionnaire. Their notes taken during task 1 tended to be more consistent and easier to compare. During each classification task, they tended to maintain attention until they located the target cluster. One technique that was particular to these users was fast pre-listening of several example sounds in a cluster, which produced a quick audio summary of that cluster. After some iterations, the different clusters had been learned and pre-listening was no longer necessary.

When looking at the results aggregated from the point of view of clusters, recognition rate was similar but with less variation ($mean = 0.48$, $sd = 0.2$). In order to understand the importance of different measures of cluster quality defined above, we built a multiple linear regression model using these measures as independent variables and the recognition rate as dependent variable. One common problem with linear regression is multicollinearity due to correlation of the independent variables, which can give misleading results. We checked the Variable Inflation Factor (VIF), for controlling multicollinearity problems, and ensured that it was below 10, which is the usually recommended threshold (Hair et al., 2010). This forced us to remove the C_{imin} and C_{emin} variables, which represent the maximal internal/external distance (minimal similarity) and thus are highly related to the corresponding mean variables. We also removed C_{emax}, as it didn't make

Table 4.2: Themes and classification criteria followed by users extracted from the questionnaire and paper notes (italics indicate verbatim quotes)

Themes	Classification criteria
Source of the sound (**16**)	· instruments (**5**): e.g., *bass* (2), *drums* (3) · onomatopoeias (**1**): e.g., *click click* (1) · *speech/vocal* (**2**) · *non-speech* (**1**) · physical source (**2**): *physical phenomena* (1) / *physical objects* (1) · electronic sounds (**4**): e.g., *synthesizer banks* (1), *base/moog* (1), *sound effect* (1) / *synthetic or futuristic sounds* (1) · *everyday sounds* (**1**)
Sound properties (**10**)	· *pitch/brightness* (**2**) / *tone* (**2**) · *length* (**3**) · dichotomies (**2**): e.g., *hard/soft* (1), *loud/low* (1) · *sound envelope* (**1**)
Experiential (**5**)	· *general feel* (**1**) · *instinct/intuition* (**2**) · *stories* (**1**) · *music experience/knowledge* (**1**)
Acoustic / visual similarity (**5**)	· sound similarity (**2**): e.g., *similar sound features* (2) · *similarity* (**1**) · visual similarity (**2**): e.g., *waveform* (1) / *waveform icons* (1)
Sound description (**3**)	· categories/tags (**3**): e.g., *my own postits* (1), *categories* (2)
Overall sound (**1**)	· *mainly the sound* (**1**)

any significant contribution to the model. Table 4.3 shows the coefficients of the model. Perhaps surprisingly, C_{imax}, the maximum similarity within the cluster, has the highest significant impact over the recognition rate, much higher than the average similarity. This means that clusters with high maximum similarity were easy to learn while clusters with low maximum similarity were difficult. In relation with the lower weight of the mean similarity, this suggests that clusters containing high-density areas allow an easier association of acoustic features with a single label, while sparse clusters, where the closest two points are not particularly close (even if the average similarity stays high) should be avoided. The rest of coefficients are more or less predictable. The size of the cluster has a small but significant negative effect, which suggests that smaller clusters are to be preferred.

Table 4.3: Regression analysis for cluster-level variables

Variable	Estimate	Std. Error	t value	p
C_{imean}	1.1229	0.6682	1.68	0.1048
C_{imax}	22.5082	9.6593	2.33	0.0278
C_{size}	-0.0009	0.0004	-2.19	0.0378
C_{emean}	-2.1158	0.8856	-2.39	0.0244
$\boldsymbol{R^2}$	0.3666			

In all, the experiment provided valuable insights with respect to the possibility of using automatic indexing of sounds into abstract lexicons defined by acoustic properties.

From the point of view of users, the result of our experiment stressed the importance of the learning phase, where most significant differences between users were observed. This suggests that interfaces for applications based on clustering could make use of a specialized interface for learning the clusters. Also, interfaces should make it easy to compare the sounds of different clusters. Finally, it seems that both references to sound sources and key acoustic properties in each cluster are common labels that users associate with the partitions.

From the point of view of the clustering algorithm, the experiment highlights the ability of clustering algorithms to find areas of high density. Still, this is not a trivial issue with heterogeneous data such as the sounds in Freesound, as preserving diversity is also important for creative applications.

4.6.2 Clustering through modularity optimization

As introduced in section 4.4, modularity-based clustering of knn graphs can be used for automatically indexing large audio databases in musical applications. In this experiment, we compare this approach to common clustering algorithms. While finding datasets labelled with hierarchical structure is difficult, the datasets described in chapter 3 allow us to evaluate the modularity clustering scheme to some extent with flat partitions (note that the *dares_scenes* dataset was obtained after completing this experiment, so it is not included). It should be noted that the idea of graph clustering is especially suited for unstructured and unevenly distributed data, where other algorithms fail. Among the available datasets, the ones extracted from community-driven sites (Freesound, Looperman and Indaba) are the most interesting in this case.

Methodology

Evaluation of clustering is always a tricky issue, as whether the groupings found by the algorithm are a good representation of the data may be very subjective. However, when labels are available, several measures can be used to compare the partition found by the algorithm with the original set of labels. These are called *external validity measures*, as opposed to *internal validity measures*, which require no labels. Internal measures are a bad choice for comparing algorithms, since many algorithms are actually based on optimization of some of these measures. The adjusted Rand index (Hubert and Arabie, 1985) is a commonly used external clustering validity measure. Given two partitions (the ground truth and the clustering solution) of the same set of elements, if a is the number of pairs of objects that are in the same set in both partitions, b the number of pairs that are in different sets in both partitions, and c, d, the number of pairs that are in different sets in one partition and in the same in the other and vice-versa, the rand index is defined as

$$R = \frac{a+b}{a+b+c+d} \tag{4.6}$$

The adjusted-for-chance version (with $E(R)$ as the expected value of R) is then:

$$AR = \frac{R - E(R)}{Max(R) - E(R)} \tag{4.7}$$

While both indices take values from zero to one, the adjusted version is typically lower.

With respect to the features, our initial experiments were based on raw filterbank features, with the idea that they would result in more easily interpretable results than standard cepstral features, in a context where subjective evaluation is important such as clustering. A cluster centroid obtained from raw filterbank features can be readily interpreted as a quantized spectrum. However, whenever distance calculations are involved, cepstral coefficients will always give a better result.

We compared modularity clustering to several classic partitional clustering algorithms: *K-means*, Self-organizing maps (*Som*), *Clara*, and Partitioning around Medioids (*Pam*) (Kaufman and Rousseeuw, 2009), as well as the partial implementation of *Chameleon* (Karypis et al., 1999) that was used in 4.6.1. All of these algorithms require the number of clusters as a parameter, so we passed the correct number of clusters. In order to get a fair comparison to the modularity-based approach, we used the *Walktrap* modularity optimization algorithm which generates a dendrogram, and obtained the desired number of clusters by cutting the dendrogram. For all datasets, we computed mean and variance of both raw filterbank features and cepstral coefficients. One classic issue with clustering algorithms is that they tend to have problems with larger feature dimensions. In order analyze the robustness of different algorithms to the *curse of dimensionality*, we tested the algorithms with a growing number of features, like in our experiments in chapter 3.

Results and discussion

Figures 4.3 and 4.4 show the results for all datasets with raw features. One general observation is that increasing the number of features rarely changes the result. Thus, clustering can be done more efficiently using the minimal amount of features. In most cases, the difference between graph-based and partitional algorithms is large. The only exception is the *indaba* dataset, which seems to be problematic for clustering, while with the SVM approach we were able to obtain high accuracies. One possible interpretation is that the supervised approach is able to focus on the specific features that allow discriminating the different instrument sounds in this dataset, while in the clustering algorithms all features have the same weight. After the graph-based approaches, the SOM (which has been previously used for exploration of audio databases) tends to perform better than other

traditional clustering algorithms. With respect to graph-based algorithms, modullarity works generally better than *Chameleon* as implemented in the CLUTO package (Karypis, 2002).

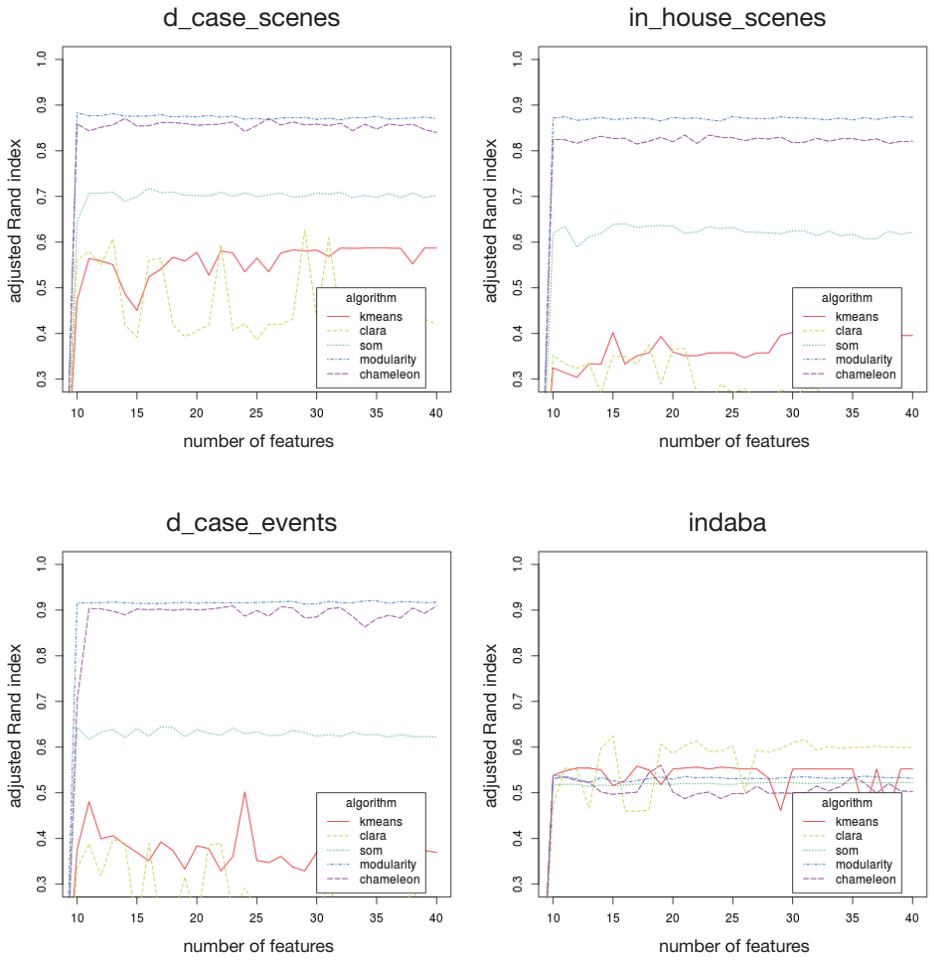


Figure 4.3: Comparison of graph-based clustering to classic clustering algorithms for the smaller datasets, as a function of the number of features

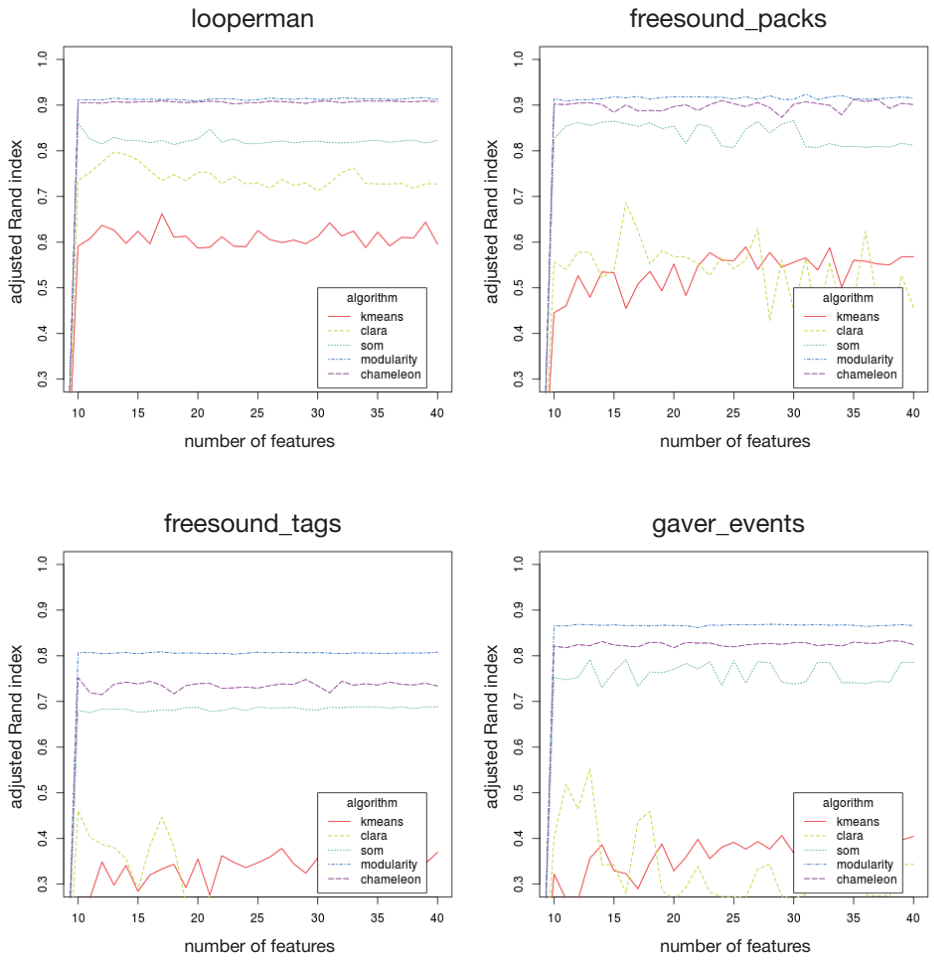


Figure 4.4: Comparison of graph-based clustering to classic clustering algorithms using raw filterbank features for the larger datasets, as a function of the number of features

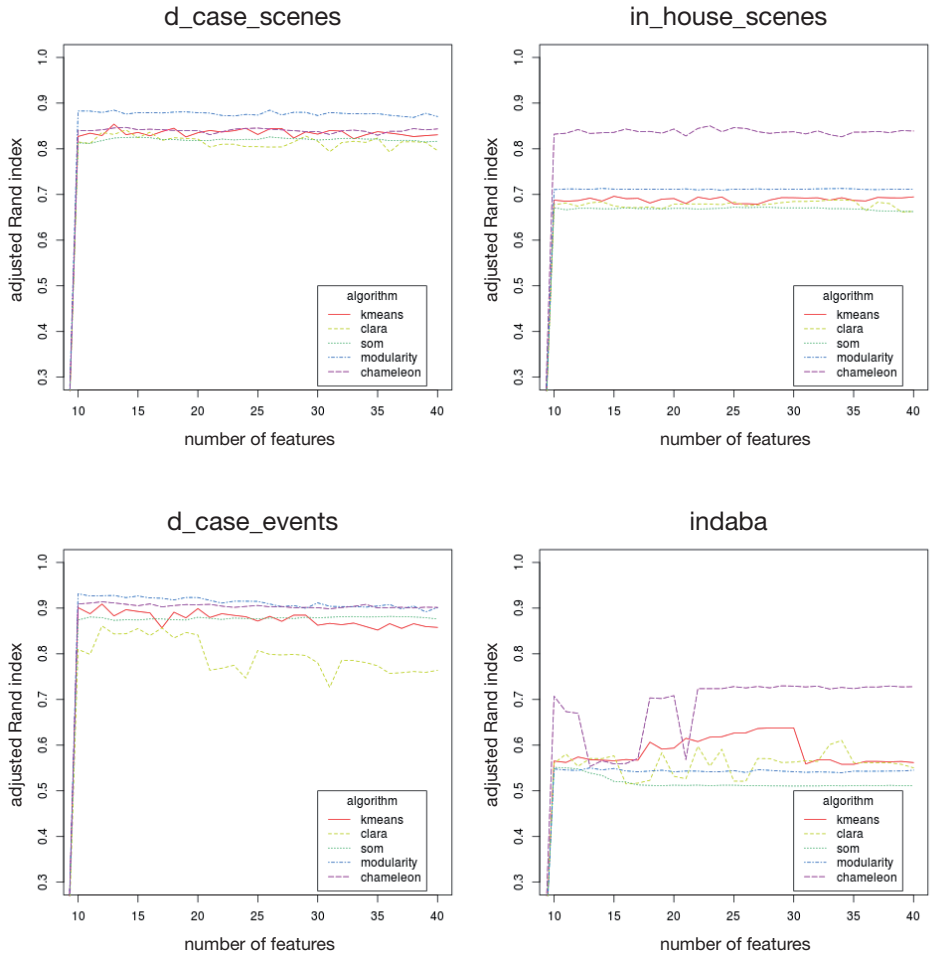


Figure 4.5: Comparison of graph-based clustering to classic clustering algorithms using cepstral coefficients for the smaller datasets, as a function of the number of features

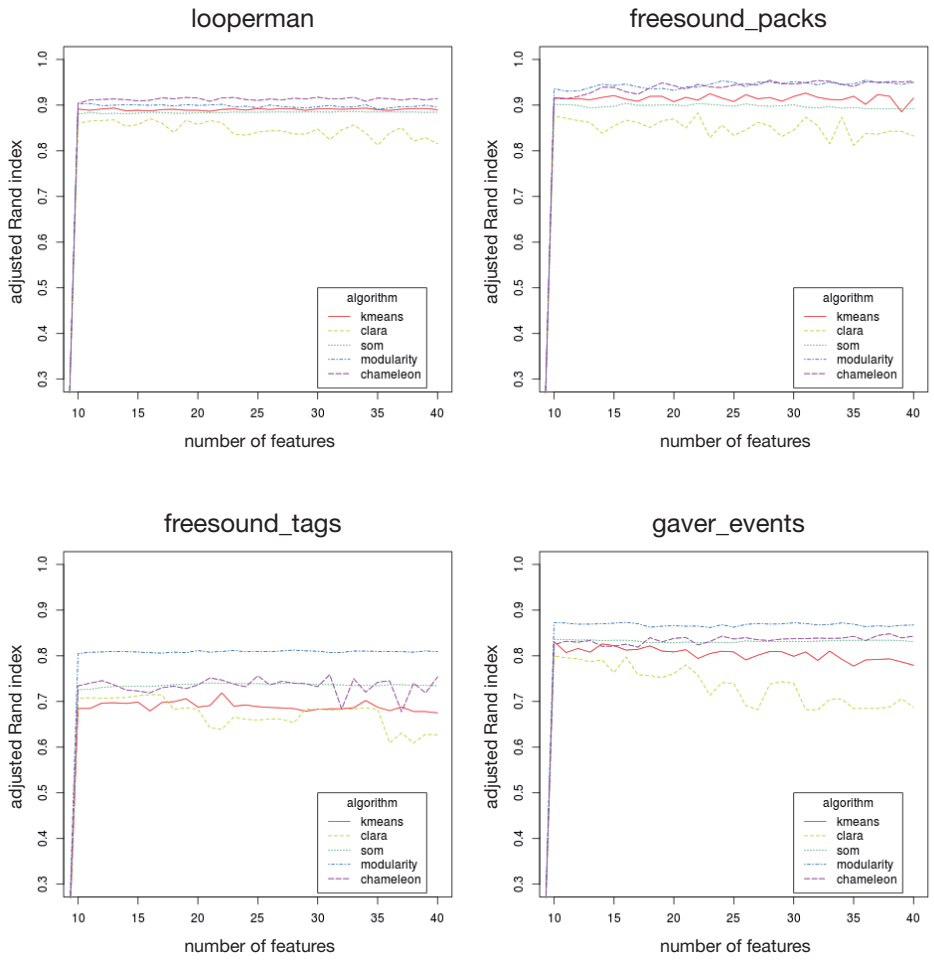


Figure 4.6: Comparison of modularity clustering to classic clustering algorithms using cepstral coefficients for the larger datasets, as a function of the number of features

With respect to using cepstral coefficients, the results are not so clear (Figures 4.5 and 4.6). Graph-based algorithms still provide a better result in most cases, but the difference in many cases is small. This can be due to some *ceiling effect*, since all algorithms generally achieve good scores. The difference in the case of the *freesound_tags* dataset (which was particularly difficult also for supervised classification) supports the idea that in the other datasets classes have an homogeneous distribution of cepstral coefficients, which makes them an easy task for clustering. On the other hand, partitional algorithms heavily rely on distance computations, which are more reliable when using cepstral coefficients, so the difference between both cases is larger. Graph-based algorithms only rely on distance computations for constructing the kNN graph, and in that case only the rank of distances (for selecting the nearest neighbors) is relevant, and not the actual distance values. Another noticeable difference can be apprecaited between *Chameleon* and modularity in the *in_house_scenes* dataset, which is composed of very diverse recordings. The difference is likely related with the fact that *Chameleon* removes some outliers (even if we tried to minimize this effect through the parameters, following the documentation, to ensure a fair comparison). Outlier removal could hence be applied to improve modularity clustering.

In all, our results indicate that graph-based clustering works better than traditional partitional algorithms for clustering audio, particularly in difficult situations such as unstructured data. The *freesound_tags* dataset is a good example of this, as the same tag may be applied in different ways by users, and distances between sounds can be different for each tag depending on the generality of the term. In addition, the *Louvain* method for modularity clustering can be used to automatically find non-binary hierarchical structures. This is uncommon in hierarchical clustering algorithms, which usually output a dendrogram. We compare this approach to a partition obtained via supervised classification on a large dataset in experiment 4.6.4.

4.6.3 Flat vs. hierarchical supervised classification

In the supervised case, we have described the use of SVM classifiers for automatic taxonomical indexing. As we have mentioned before, SVMs are already ubiquitous for audio classification, so it is not necessary to evaluate them for audio classification in general. For application to taxonomical classification, we have mentioned two straightforward approaches: we may

train a hierarchy of multi-class classifiers, or we may just train one single multi-class classifier at the lowest level, and infer the higher levels from the taxonomy itself. In this experiment we compared both approaches with the gaver taxonomy described in section 4.2.

Methodology

The methodology is similar to the experiments of chapter 3. In this case, to compare both approaches, we randomly partitioned the database into a training and a test set (80% and 20% respectively) and repeated the process 10 times. In each round, we trained either the flat classifier or the set of hierarchical classifiers with the training set, and computed the predictions for the test set. We extracted mean, variance and *RQA* features, computed from *GFCC* coefficients, from the *gaver* dataset, in this case taking into account the two levels of the taxonomy for the labels. The flat classifier was trained using the *one-to-one* approach with the 11 base classes of the taxonomy. For the hierarchical classification approach, we trained one *one-to-one* classifier for the three kinds of material (*solid*, *liquid* and *gas*), and then another classifier for each one, using only the training examples of the corresponding material. For each test sound in the hierarchical approach, we first predicted the top class with the first classifier, and then applied the second-level classifier depending on the result. Because of the random partition, the number of examples per class was not balanced. In both cases, we computed the average F-measure, which is more appropriate than accuracy for dealing with unbalanced datasets. We averaged the F-measure between classes, and again for each round.

Results and discussion

Figure 4.7 shows the *F-measure* for the flat and the hierarchical approach. Both classifiers tend to choose the correct class about 70% of the times, which is not a bad result considering the generality of the concept of gaver's taxonomy and the number of classes, which implies a baseline of 9% for a random classifier. Particularly in the case of music creation applications, some the errors may be acceptable as they are still based on content descriptors. A more detailed view is provided by (row-normalized) confusion matrices for both levels (Figure 4.8). Here, numbers indicate the fraction of correctly classified sounds for each class. In the case of the top level, both approaches perform very well, and the flat approach (where the top level label is inferred from the lower level label) can perform even better than di-

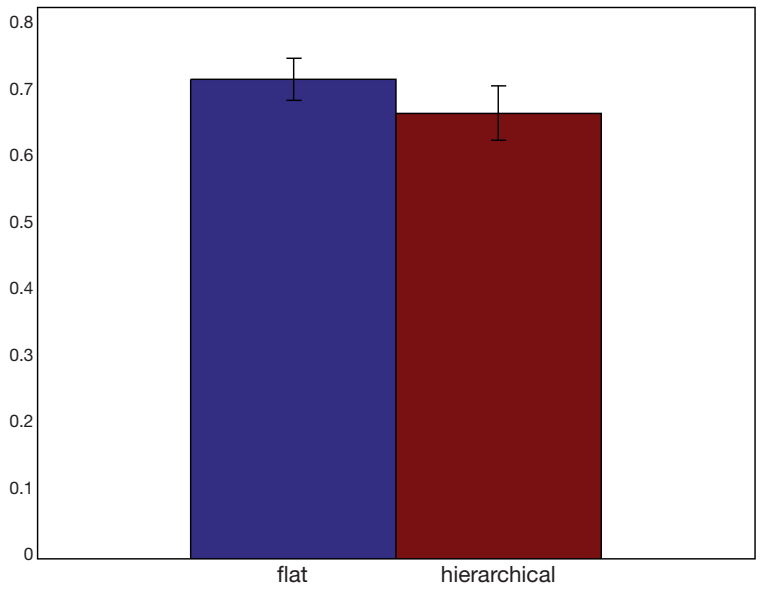


Figure 4.7: Flat vs. hierarchical classification (*F-measure*)

rect classification of materials. In general, the performance of a hierarchical approach will depend on the coherence of the top-level concepts with respect to audio features, which may be complicated for very general concepts. At the lower level, the hierarchical approach makes more errors, particularly on some difficult categories (*deformation*, *splash*), and the errors are spread among different classes. The errors of the flat approach are also related to the same difficult classes but are more localized and scarce. In general, flat classification performs better, which can be attributed to the propagation of errors across the levels in the case of the hierarchical approach.

4.6.4 Supervised vs. unsupervised indexing

In this chapter, we have introduced two methods for automatical taxonomical organization of sounds, which can be applied to the case of unstructured audio databases for supporting music creation. Unsupervised and unsupervised approaches have very different implications and uses. The main difference is that supervised methods require some training examples that may require a significant amount of effort, while unsupervised methods rely on the distribution already present in data. Thus, supervised approaches

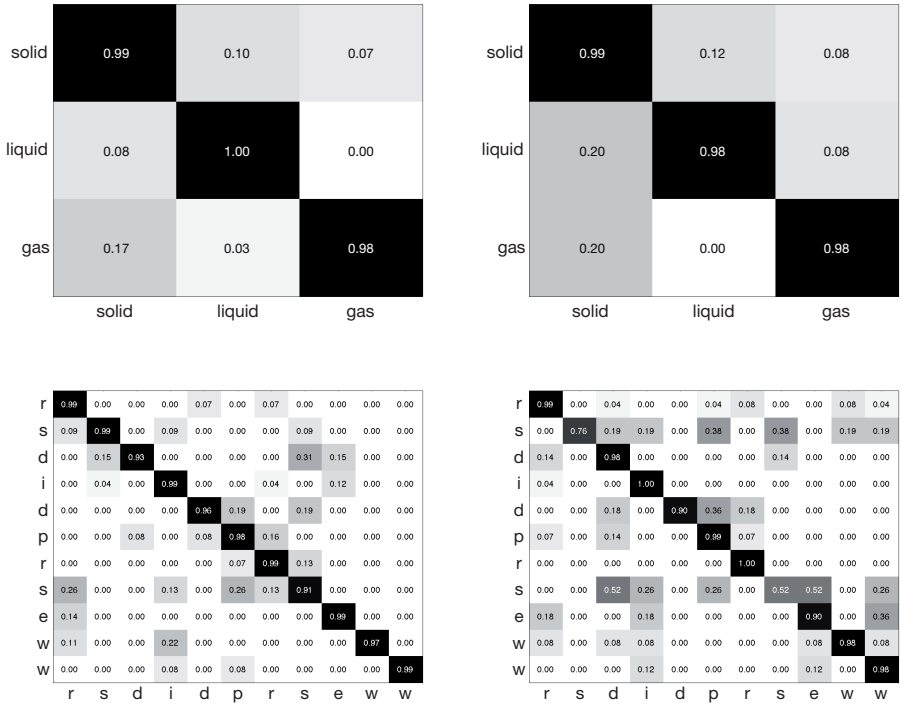


Figure 4.8: Row-normalized confusion matrices

can be seen as *top-down*, in the sense that somebody (typically an expert) has to devise the taxonomy that will be used to index sounds. Unsupervised approaches can be seen as *bottom-up* and, particularly in the case of community-driven databases, reflect groupings that emerge from different views, so in a way they depend on self-organization of the community. In order to compare both approaches for the case of unstructured audio data, we performed a user experiment, where we asked users to find sounds using a visual representation of a taxonomy without labels. The idea was to test the usefulness of the supervised and the unsupervised indexing approaches for interacting with large audio databases in an intuitive way, for classes of sounds that are derived from content descriptors, and not from semantic descriptions. Our hypothesis is that bottom-up clustering based on modularity optimization can at least match the results of a top-down classification approach for indexing sounds in unstructured databases, without the need of labelling training examples. The experiment focused on sound events,

extracted from the Freesound database.

Methodology

We analyzed all of the Freesound database, and identified all sounds containing a single event using the HFC_{fb} onset detection function described in chapter 3. However, we did not segment the original sounds but retained just the ones containing a single event. This resulted in a database of 28.295 sounds, which were analyzed in the same way as in the previous experiment. We then used the *gaver* taxonomy dataset to train a flat classifier, and indexed all sounds according to the *gaver* taxonomy. The classifier was trained with probability estimates, which allowed us to know, for each sound, the probability that it belongs to a given category in the taxonomy. In order to obtain a comparable taxonomy using the modularity clustering approach, we used the *Louvain* algorithm to obtain a multi-level partition, and then discarded the nodes containing less sounds, allowing a maximum of 4 children for each taxonomy node.

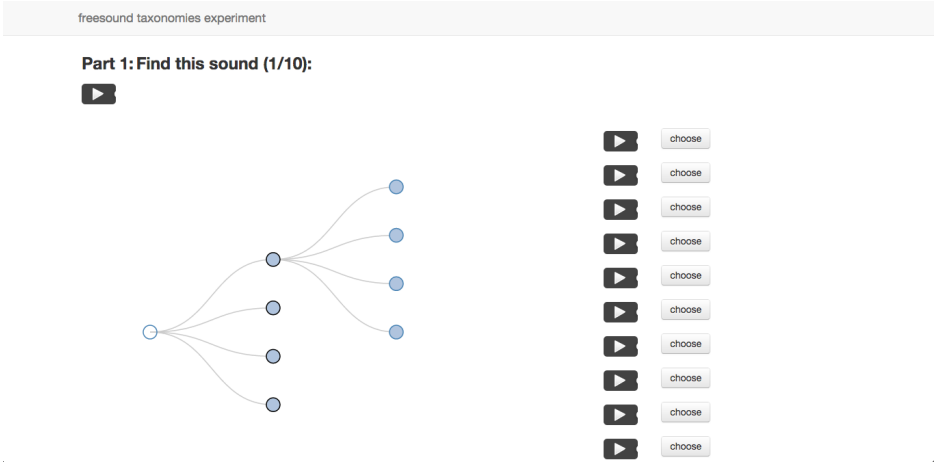


Figure 4.9: Screenshot of the prototype

We developed a web prototype where the taxonomies were displayed graphically (Figure 4.9). Nodes in the taxonomy could be unfolded and folded to display or hide their children. For each node in the taxonomy, 10 example sounds were selected as "previews". The examples were selected at the lowest level of the taxonomy and then, for higher levels, examples were sampled uniformly from children nodes. In the case of the SVM-based approach, examples were the sounds with maximum probability estimates. In

the unsupervised approach, we computed the centroid of each cluster and sorted the sounds according to their distance to the centroid. Rolling the mouse over the taxonomy nodes allowed the user to listen to the examples for each class, in the same fast-skipping style that some participants used in the first experiment. When a final node (with no children) was selected, a list of 20 sounds corresponding to that node was presented. These were sorted using the same criteria as the examples described above. The protocol was as follows: each user was presented a series of 20 challenges, 10 for each taxonomy (in random order). For each challenge, a target sound was presented, and the task was to locate the sound among the final lists of 20 sounds by navigating through the tree and playing/listening to sounds. Sounds could be played as many times as needed. The taxonomy was meant to help in the process of finding a sound in a sample of the database (the sample size was adapted to so that the user could do the task in about 20 minutes). In order to keep the duration of the experiment to a manageable range, users were instructed to find the most similar sound they could find to the target sound, so that they would not spend to much time finding the exact sound.

With this set-up, we measured several variables corresponding to the cost of navigating the taxonomy: the number of visited nodes (VN), the number of unfolded nodes (UN), the number of times that the target sound was played (TP), and the number of candidates played (CP). We also computed the distance between the target and the chosen sound in the taxonomy (as the number of common ancestors, TD) and the content-based distance between both (CD). We then performed several ANOVA tests using these measures as dependent variables, in order to see if their variances could be explained significantly depending on the type of taxonomy. Perhaps more than the cost of navigation, we were interested in the users ability to learn the taxonomical organization, in the light of the findings in experiment 4.6.1. In order to understand how users were able to learn each taxonomy, we divided the 10 trials in two groups and computed the mean of each variable for each group of 5 trials. For each user and taxonomy, we computed the difference between the first and the second group of trials. The hypothesis was that users would require each time less visits to the different nodes in the taxonomy in order to find the target sound (with some fluctuation given the different relative difficulties of the random target sounds). We also analyzed the user as a potentially relevant factor, and whether the exact sound had been found or not (which can help understanding why a user spends a lot of time browsing the taxonomy). The experiment also

included a final questionnaire with some demographic information and a question about what part of the experiment was found to be easier.

After a pilot test, the experiment was completed by 16 participants, only one of them female. Ages ranged from 20 to 50. Most of them (12) had more than 5 years of music training, and the same number considered themselves experts in music technology.

Results and discussion

Figure 4.10 shows box plots to summarize the distribution of each of the aforementioned variables with respect to each of the taxonomies. The values for the corresponding ANOVA tests are shown in table 4.4. Clearly, there is a significant effect of the taxonomy over the navigation costs, with the supervised approach requiring less user interaction, which indicates that users spent less time navigating the taxonomy. Also, the number of correctly identified sounds was higher for the supervised approach. However, this approach also required significantly more time to be spent on the list of candidate files. This is precisely what the taxonomical indexing tries to avoid: to have users listen to the sounds one by one. Instead, they are expected to learn the taxonomy in order to find sounds. After listening to the sounds selected by each approach, it seems clear that the unsupervised approach tends to find clusters of very similar sounds. This seems to be the reason that users allowed themselves to settle on a similar sound instead of the original sound. Hence, with respect to the content-based distance, the sounds selected using the unsupervised taxonomy were in the end significantly closer to the original. Also, with respect to the direct question of which part was easier, 44% of users opted for the unsupervised taxonomy, 25% for the supervised taxonomy and 31% found them equally difficult (note that the order of the parts was randomized, users were asked if they had found part 1 or part 2 easier). Thus, the cost of listening to the candidate sounds seems to be what relates to the perception of difficulty of the task.

Figure 4.11 and the corresponding table (4.5) show the box plots and ANOVA tests for the difference between the first and the second half of the trials of each taxonomy, with respect to all of the cost variables. In most cases, the type of taxonomy is shown to be relevant for explaining the variance in the cost variables, with a higher weight than the user variable. In general, the unsupervised approach tends to have positive values, indicating that the costs decreased between the first and the second part of each round for each user. Thus, the results indicate that the longer time

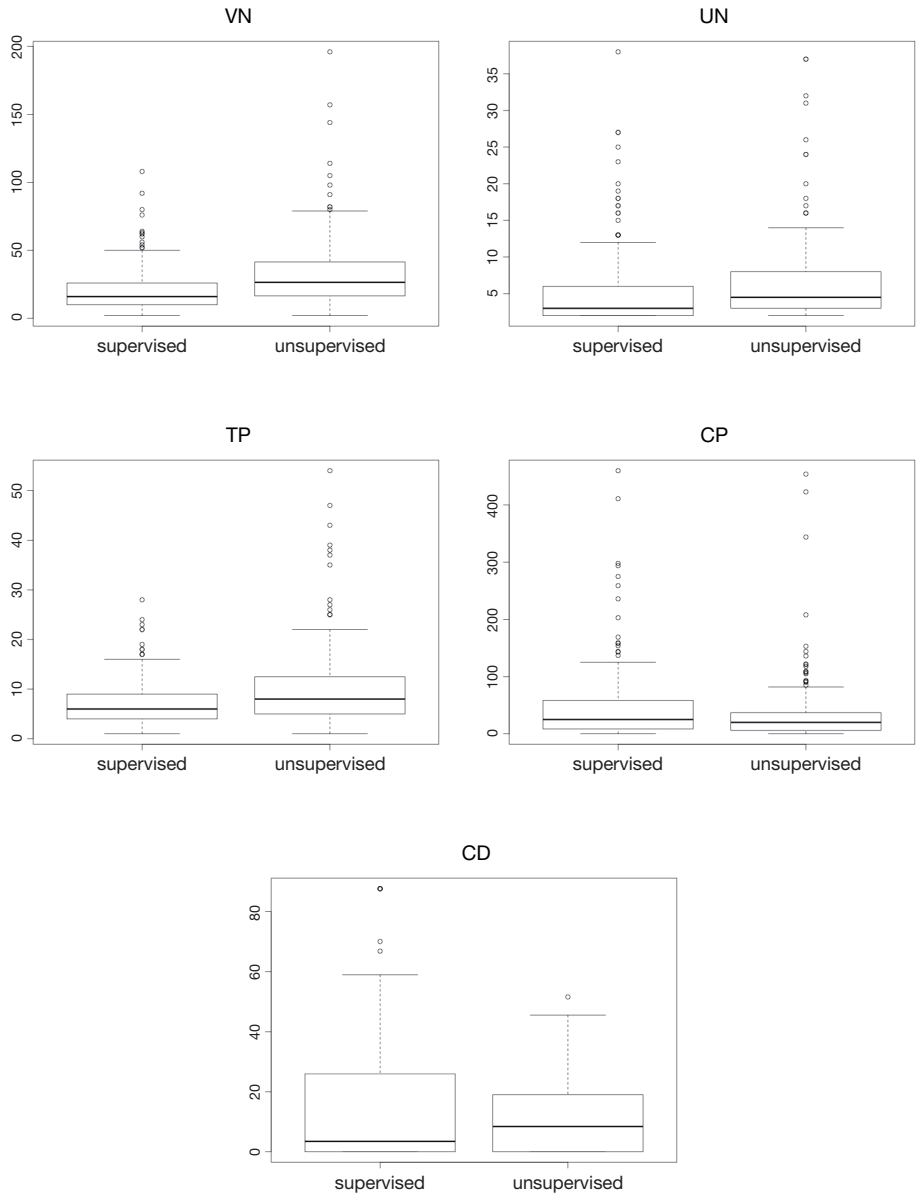


Figure 4.10: Box plots for the experiment variables with respect to the two taxonomies

Table 4.4: Results of the ANOVA tests with the cost variables and the binary *found* variable

	Df	Sum Sq	Mean Sq	F value	p
taxonomy	1	1.51	1.51	6.37	0.0121
user	1	1.38	1.38	5.82	0.0164
Residuals	317	75.30	0.24		
					FOUND
taxonomy	1	12115.50	12115.50	22.60	0.0000
user	1	5813.63	5813.63	10.84	0.0011
found	1	806.02	806.02	1.50	0.2211
Residuals	316	169416.34	536.13		
					VN
taxonomy	1	70.31	70.31	1.88	0.1713
user	1	9.64	9.64	0.26	0.6120
found	1	11.80	11.80	0.32	0.5746
Residuals	316	11815.73	37.39		
					UN
taxonomy	1	784.38	784.38	15.55	0.0001
user	1	611.10	611.10	12.11	0.0006
found	1	15.44	15.44	0.31	0.5805
Residuals	316	15941.33	50.45		
					TP
taxonomy	1	14311.25	14311.25	3.10	0.0793
user	1	4266.03	4266.03	0.92	0.3373
found	1	8868.18	8868.18	1.92	0.1668
Residuals	316	1459632.02	4619.09		
					CP
taxonomy	1	63.90	63.90	96.81	0.0000
user	1	4.43	4.43	6.71	0.0101
found	1	129.58	129.58	196.30	0.0000
Residuals	316	208.59	0.66		
					TD
taxonomy	1	1710.93	1710.93	9.79	0.0019
user	1	3038.52	3038.52	17.38	0.0000
found	1	46960.03	46960.03	268.65	0.0000
Residuals	316	55236.65	174.80		
					CD

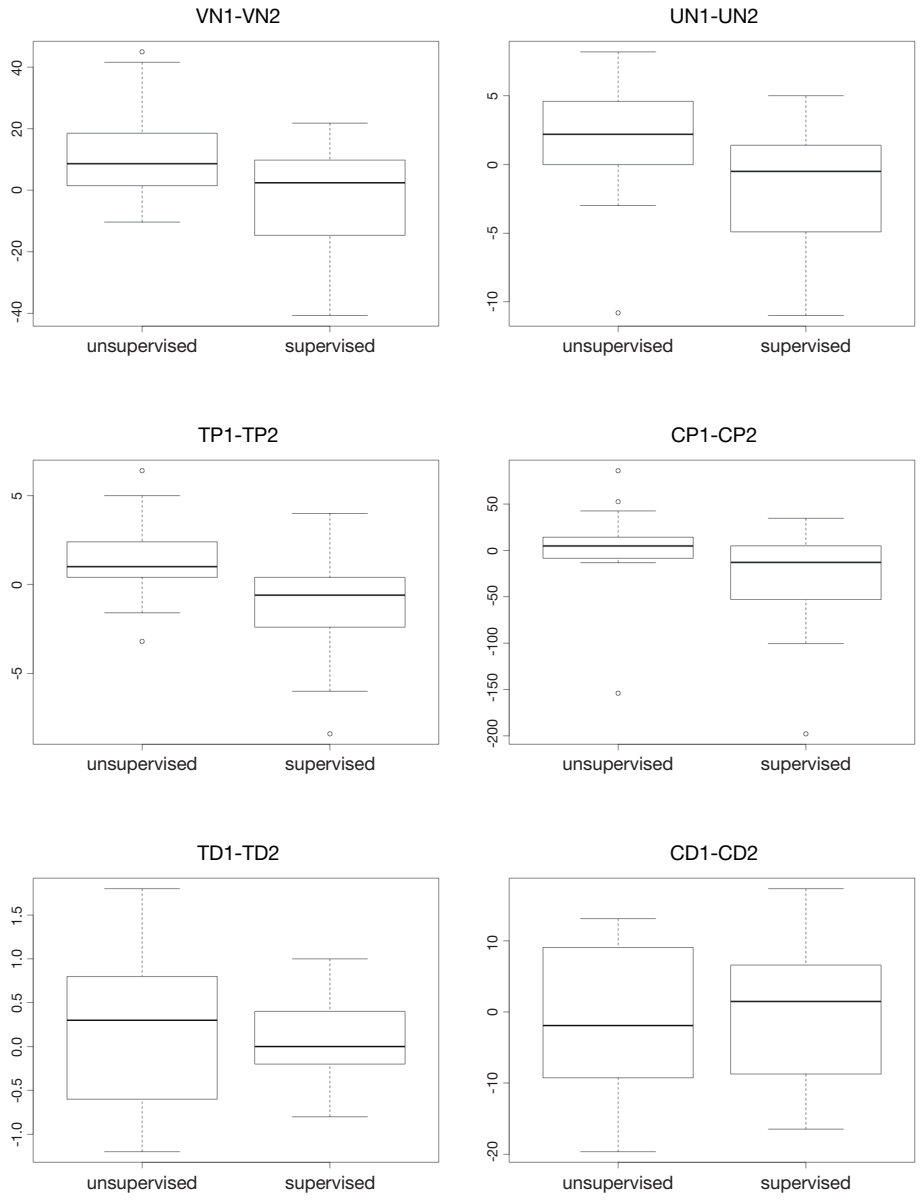


Figure 4.11: Box plots for the relative change in experiment variables with respect to the two taxonomies

Table 4.5: Results of the ANOVA tests with the relative change in variables

	Df	Sum Sq	Mean Sq	F value	p
taxonomy	1	2051.20	2051.20	6.70	0.0148
Residuals	30	9190.01	306.33		
					VN1-VN2
taxonomy	1	84.50	84.50	4.07	0.0526
Residuals	30	622.46	20.75		
					UN1-UN2
taxonomy	1	52.53	52.53	6.56	0.0157
Residuals	30	240.32	8.01		
					TP1-TP2
taxonomy	1	8450.00	8450.00	2.91	0.0986
Residuals	30	87218.56	2907.29		
					CP1-CP2
taxonomy	1	0.06	0.06	0.11	0.7420
Residuals	30	16.65	0.55		
					TD1-TD2
taxonomy	1	0.16	0.16	0.00	0.9688
Residuals	30	3060.83	102.03		
					CD1-CD2

spent pre-listening to the taxonomy examples was associated to a better understanding of the taxonomy, which resulted in less time spent playing the candidate examples. Our general interpretation of the experiment is that a top-down approach provides easily understandable concepts, which require less time to understand, but leave no room for improvement. Part of this success can be attributed to the strategy of selecting the best examples with the probability estimates of the SVM classifier. However when reaching down to the last level, sounds are less uniform and users need to play more sounds in order to find what they are looking for. Contrastingly, the bottom-up approach does not require labeling training examples and is based on the existing groupings in the data. Interacting with this type of taxonomical organization may require some learning, and the choice of appropriate strategies for selecting examples will be critical. However in the

end searching sounds using the unsupervised approach is perceived as an easier task, as when the final sounds are reached, the groupings are more consistent.

4.7 Conclusions

In this chapter we have analyzed automatic organization of sounds into taxonomies, using machine learning methods with the low-level representations described in the previous chapter. The main ways to obtain such taxonomies are either top-down classification, based on predefined categories, or bottom-up clustering based on the data distribution. We have presented a novel method for clustering audio by modularity optimizattion on kNN graphs. Graph-based clustering helps adapting to different densities found in user-driven databases. Many algorithms based on modularity optimization exist, some of which can deal with large scales such as currently found in online audio databases. This provides a flexible framework that can estimate the number of clusters, find a partition for a user-specified number, or even compute a non-binary hierarchical partition. We have also shown that modularity-based clustering tends to perform better than traditional partitional clustering algorithms. With respect to supervised classification, we have analyzed the use of conventional SVM classifiers with a general environmental sound taxonomy. We have seen that direct classification of the last level in the taxonomy tends to perform slightly better than using different classifiers at each level. Finally, we have conducted a user experiment comparing top-down and bottom-up methods for generating taxonomies. The results indicate that top-down classification, as described in this chapter, helps in finding representative sounds that make taxonomies understandable. However, at lower levels they do not provide consistent groupings as clustering algorithms for large databases. Taxonomies based on clustering, again using our proposed method, are more easily learnt so they seem more appropiate for long-term users.

In all, the techniques described in this chapter should be useful for interacting with large databases, affording a different interaction style than text search or Query by Example (QbE). In this case, users can learn sound taxonomies that are created without the need of manually annotating all the sounds, and find interesting sounds by memory. However, there is still work to do in order to understand how this kind of taxonomies could work in practice for very large datasets. Particularly of interest is choosing appropriate representatives for clustering-based taxonomies. Bottom-up and

top-down approaches could be combined in order to otain the benefits of each method. In addition to navigating databases, these taxonomies can also be used for obtaining symbolic representations of music based on audio samples. We analyze this possibility in chapter 5.

Chapter 5

Representing music as work in progress

5.1 Introduction

In chapter 1, we have commented on the opportunities that shared storage opens for music creation based on audio. Practically any audible sound can be captured into an audio recording, and thus audio samples can potentially be used in any musical genre, at the expense of flexibility. Most music produced nowadays involves at some point editing digital audio, so the boundaries between specifically audio-based musical practices and other kinds of music have become fuzzy.

In this thesis we analyze audio-based music creation on the web. In this context, separation of musical structure from the audio samples allows the use of hosting services and shared databases for the audio data, while musical structure can be represented and exchanged using text markup formats such as XML or JSON (Figure 5.1). In creative activities, music structure can be typically stored in lightweight documents that may change frequently, and transmitted through established text communication channels, such as standard web technologies or email, including environments with constrained or expensive bandwidth (e.g. mobile connections). Multiple revisions of text documents can also be managed using existing revision control systems, including WebDAV (Whitehead Jr and Wiggins, 1998). The use of version control for music collaboration has been investigated in the CODES project (Miletto et al., 2009). Version control is also obviously used often for music represented as computer programs. In this chapter, we propose

a music representation based on text markup, where each text document may make reference to a number of bigger sized audio files. Transmission and local caching of these files can be managed independently between each participant and the remote location through standard web technologies and services, which avoids the need of potentially complex specialized p2p tools for the synchronization of audio collections among different participants.

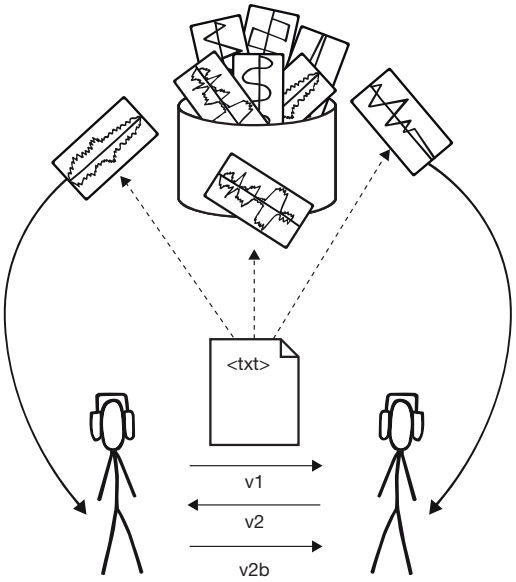


Figure 5.1: Markup plus audio representation

In recent times, JSON has generally replaced XML as the *de facto* representation for exchanging data in web applications and services, although without the emphasis on interoperable standards. On the other hand MusicXML has become a successful standard for applications using classical western music notation, including online tools such as Noteflight[1]. However, there are many musical practices that cannot be represented using classic western notation. In this chapter we propose a representation for audio-based music that can be easily encoded as JSON or XML, and affords the definition of simple grammars for organizing collaboration. We describe a proof-of-concept experiment where this representation was used in a collaborative music creation prototype. The proposed approach allows

[1] `http://www.noteflight.com`

the identification of common patterns in user-submitted musical artifacts based on audio samples.

Part of this chapter was published in (Roma and Herrera, 2010b) and (Roma and Herrera, 2013).

5.2 Grammar framework

For the case of collaborative music composition, grammars provide a suitable framework for sharing, reusing and assembling parts of music compositions. In this sense, a grammar can serve as the formal representation of nested musical fragments, and eventually support the metaphor of a common language. As an example, a group of users or a program could establish some rules with respect to music compositions, such as defining a fixed set of instruments for all pieces, or some common structural elements. This could be interpreted as a set of grammar rules for music creation. From the point of view of analysis, grammars can be used for computational modeling of the style of different participants or groups in collaborative applications. From the perspective of generation, they can serve to facilitate creativity by producing new combinations of sounds, or to assist the search of suitable sounds for a given musical context. Thus, while the main concepts of formal grammars can be found in any textbook, we now provide a summary in order to introduce the idea of using grammars for audio-based online music creation.

5.2.1 Formal grammars

A formal grammar is usually defined as a 4-tuple (V, Σ, S, P), where:

Σ is a terminal alphabet, a set of symbols that are used to form sentences in the language. In common music notation, terminals could be note pitches, note durations or chords, while in sample-based music they can be sound objects (Roads, 1978), or specific sound object classes. V is an alphabet of non-terminal symbols or variables. Variables represent strings of terminal or non-terminal symbols in intermediate stages of the generation. In a music composition process, variables can be used, for example, to represent groups of terminals that are often used together, such as chords or rhythmic patterns. S is the start symbol, a special variable that is used to begin the generation process. P is a set of production rules that allow a given string to be replaced by another string. For example they can specify how a part of a musical piece can be replaced by its subparts.

In summary, a formal grammar can be described as a set of rules that rewrite a string of symbols with another one. For example the rule A →AB defines that a string composed of the symbol "A" can be replaced (rewritten) by the string "AB". This rule can be used to generate the string "AB" in presence of string "A". Thus, the rule could be applied recursively to "AB" to produce "AAB", and so on. Also, it can give one possible explanation of how the string "AB" (or "AAB" for that matter) was produced in a given language.

Intuitively, grammars can be understood as a formal way to specify structural groupings of a language. For example we can state that all sentences in a natural language are composed of a "noun"' sub-sentence and a "verb" sub-sentence, and then define a rule that describes this decomposition using abstract symbols (say S → NV). The application to common structures such as typical pop song structures is straightforward.

5.2.2 Graph Grammars

One issue of music grammars that is not covered by linguistics or formal languages literature is parallelism (Baffioni et al., 1984). Some of the experiments with grammars for music generation in the 70s and 80s involved the use of parallel rules, where two parallel tokens are meant to start at the same time (Roads, 1978; Holtzman, 1980). However, parallel rules in string grammars introduce some ambiguity. For example if we have a musical sequence "AB" and a parallel rewriting rule "A → D/E" (meaning that D and E start at the same time), it is not clear, upon replacement of A, if B will follow after D or after E. Graph grammars provide a general framework that allow us to deal explicitly with sequential and parallel structures.

Graph grammars were introduced by Pfaltz and Rosenfeld in the late 1960s (Pfaltz and Rosenfeld, 1969) as an extension of traditional grammars to languages of directed graphs. A directed graph is defined as a tuple (N,E) where N is a set of nodes and E a set of edges that connect nodes in a certain direction. Clearly, strings are a class of directed graphs where symbols are nodes and edges define the sequence of symbols. In this sense, edges of a string define a total order relation. For acyclic graphs, the set of edges defines a partial order relation on the nodes, which allowed the generalization of string grammars to acyclic directed graphs.

A graph grammar can be defined in similar terms to string grammars. However, graph rewriting productions are more complex than string rewriting productions as they have to define how to connect the result of the pro-

duction to the enclosing graph. Thus, productions are defined as triples (α, β, E) where α is the (sub)graph to be replaced and β is the replacement, while E defines the embedding of β in the host graph. Graph grammars can be categorized in the same way as string grammars. In addition, graph grammars can focus in nodes or edges in different ways. For example, node replacement grammars (Engelfriet and Rozenberg, 1997) are context-free graph grammars where the left hand of each production is restricted to a single node.

Development of graph grammars has continued over the years both at a theoretical and at a practical level, fostered by applications in very diverse fields such as image recognition or graphical languages for engineering (Andries et al., 1999). The extension of strings to graphs seems naturally suited for music representation by explicitly dealing with parallelism. However, experiments with graph grammars for music are rare in the literature. Some works (Holder and Cook, 2009; Madsen and Jørgensen, 2003) have used them for mining classical music scores represented as graphs with multiple possible connections between consecutive notes. Since these connections are not specified in the score, accounting for all potential connections bears a complexity that may be avoided in the context of music creation.

5.3 Tree representation

For the case of online creation, graphs can be used to represent music structures where nodes represent sounds from a shared database. We name these musical artifacts as "sample patches". A simple representation for such patches is a rooted tree. A rooted tree can be defined as a directed acyclic graph with a root node where there is a unique path from the root node to any node. Vertical rooted trees are commonly used to represent monophonic melodies or rhythms, as well as music structure. An example of music representation with a vertical hierarchy is MusicXML (Good, 2001). On the other hand, a horizontal tree allows representing multiple tracks and can be used to encode music in real-time, when the end of the piece is not known, without backtracking. In this chapter, we explore the use of horizontal rooted trees for audio-based music representation. Figure 5.2 shows two example representations for a drum pattern. Figure 5.3 shows an example of a vertical tree to represent the drum/snare pattern for comparison.

The use of horizontal rooted trees affords an intuitive playback model but

implies two main limitations. On one hand, no cycles can exist in the graph. This means that edges define a partial order relation, which allows them to represent unambiguous time sequences. Also acyclic graphs are contractable (Pfaltz and Rosenfeld, 1969), which allows the definition of grammar expansion rules. One limitation of this approach is that it is quite common in music to use cyclic structures. This means that musical cycles should be understood as repetitions of a single node, potentially representing a contracted subgraph (Figure 5.4). A second restriction of this representation is that a given node can only be the target of one edge. Two edges arriving at the same target would imply the scheduling of the same node (and all the following structure) at two different moments in time, which has the effect of creating multiple audible tracks from the same graph specification, and breaking the intuition of the temporal sequence of the representation. Given these restrictions, reproducing music encoded in the graph can be thought as a tape head spawning copies of itself at the bifurcations of the tree. The main issue is how to generate the links between samples, since in traditional sequencer representations, links are not specified. The two alternatives are either to get the user to specify the links as part of the music creation process, or to create them in the interface according to some predefined rule. For example in a drum machine interface, the rule could be to generate always a string for each type of sound (i.e example *a* in Figure 5.2).

5.4 Lexical generality level

Historically, the interest in formal grammars for music composition was influenced by their success with natural language. In natural languages, the meaning of words has generally nothing to do with their written or phonetic representation, it is defined by convention. The same happens with many musical symbols such as notes, which refer to a discretized accoustic property in an abstract manner. This parallelism in the use and articulation of discrete symbol systems has been related to a more general principle of self-diverifying systems, which enables music creativity (Merker, 2002).

The concept of a *lexical map* between terminal tokens of a grammar and actual sound objects was investigated by Roads (1978). Roads defined three general forms of lexical mapping: arbitrary, injective and polymorphic. In the first case, a symbol was arbitrarily mapped to a sound object. The two other types required a lexicon of sound objects that is grouped according to some acoustic features. In injective mapping each terminal mapped to

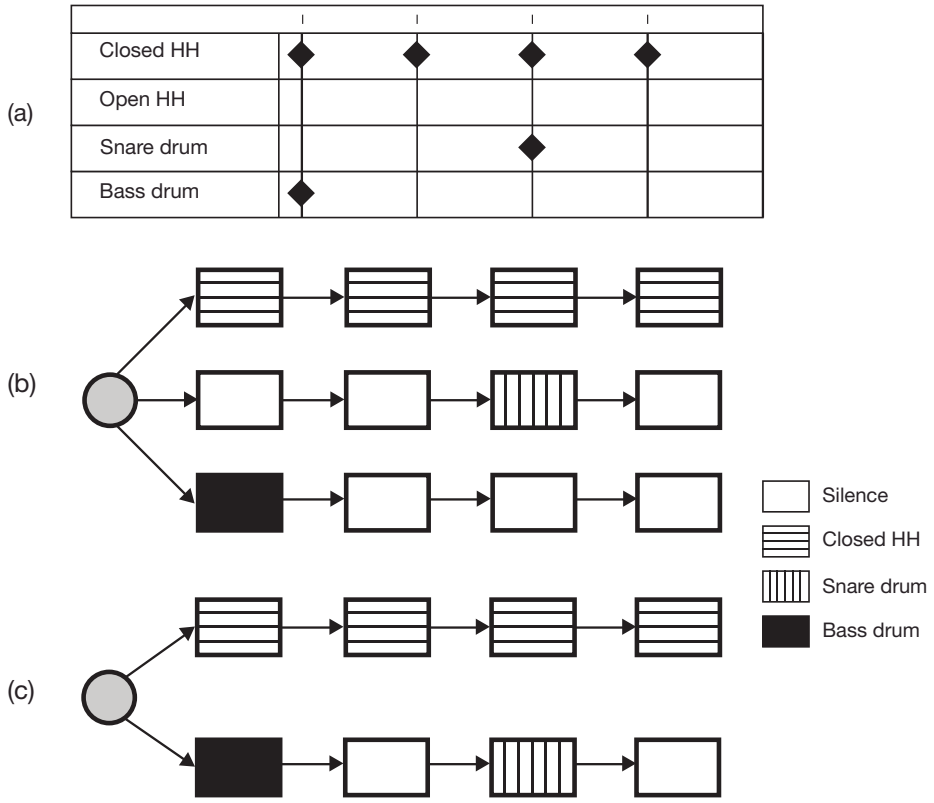


Figure 5.2: Two representations for the same drum pattern using horizontal rooted trees

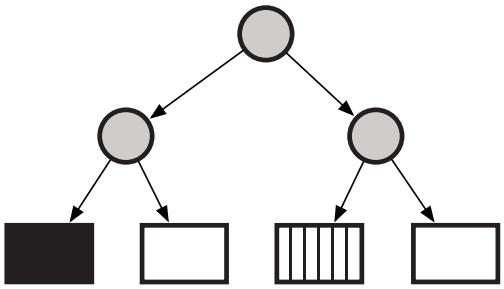


Figure 5.3: Bass/snare drum pattern represented as a vertical tree

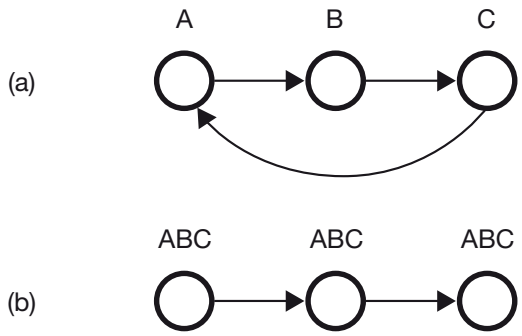


Figure 5.4: Loop represented as repetitions of the contracted subgraph

one sound from the ordered lexicon. Polymorphic mappings were regarded as a complex situation (equivalent to context-sensitive rules), allowing one-to-many and many-to-one correspondences with the lexicon.

Using current content-based audio analysis techniques, these ideas can be applied for symbolic representation of audio-based music. In this sense we may consider the alphabet of symbols used for grammar rules to be a partition of a collection of sound objects, so that all elements of the database are mapped to some symbol of the alphabet. If the partition is hard (i.e., each sound belongs to only one group), the mapping is equivalent to a polymorphic "one-to-many" mapping in Roads' terminology ("injective" mapping being a particular case when there is one sound per group). Soft partitions, such as fuzzy or overlapping partitions would pose additional problems. For example mining patterns in musical graphs where each node can have more than one label would result in a combinatorial explosion. A perhaps more sensible approach is to consider different perceptual facets (e.g. pitch, timbre, loudness) where hard partitions can be used to obtain a discrete symbol system, and use different grammars for each facet.

As we have seen in chapter 4, one way of dealing with large databases is to automatically compute taxonomies from content descriptors. This means that the same sound will belong to different (hard) classes depending on the level of the taxonomy. Thus, for dealing with large audio databases, it is convenient to define different *lexical generality levels*, corresponding to sets of nodes that are considered to be at the same height in the taxonomy. At the top level, all sounds would be labelled with the same label, which defines purely structural patterns. As we descend in the taxonomy, we obtain larger

alphabets, and more specific patterns. This approach allows the extraction of patterns with different levels of detail.

5.5 Finding patterns

We have described how grammars are commonly used both to parse or analyze the structure of musical compositions and to generate new compositions. In the context of online music making, grammar rules can be used to analyze and model the style of different users or groups, and suggest potential interesting combinations of sounds. Grammars can also be seen as an agreement between remote collaborators working in different parts of a musical work, and they can be implicitly or explicitly enforced or supported through music creation interfaces.

A graph grammar rewriting rule defines that a node of the *host* graph can be replaced by a subgraph. In Node Label Controlled (NLC) grammars (Engelfriet and Rozenberg, 1997) embedding rules are defined globally for all rules as relations among nodes with some specific labels. This mechanism can be used as a simple approach to define musical grammars for collaboration. For example, one may concede that any music fragment (represented as a horizontal tree) has a starting point and an ending point, represented by virtual terminating nodes. A node replacement grammar can then be defined so that when a node is replaced by a graph, the starting point inherits incoming edges of the replaced node, and the ending point inherits its outgoing edges. The ending node does not need be the one that ends last, but the node that conceptually ends a given pattern. This strategy does not allow maintaining parallel connections among "tracks" in each patch. Still, it can be argued that the need of such parallel connections implies the need of separate rules.

In this simplified setting, grammar expansion rules consist in trees that can be collapsed and contained in other trees. There are two main alternatives for defining such rules: user-defined groupings, and automatic groupings. User-defined groupings allow sharing sample patches for embedding into other patches. This can be done by defining virtual start and end nodes for the embedding (Figure 5.5). In addition, automatic groupings can be found inside patches for analysis purposes.

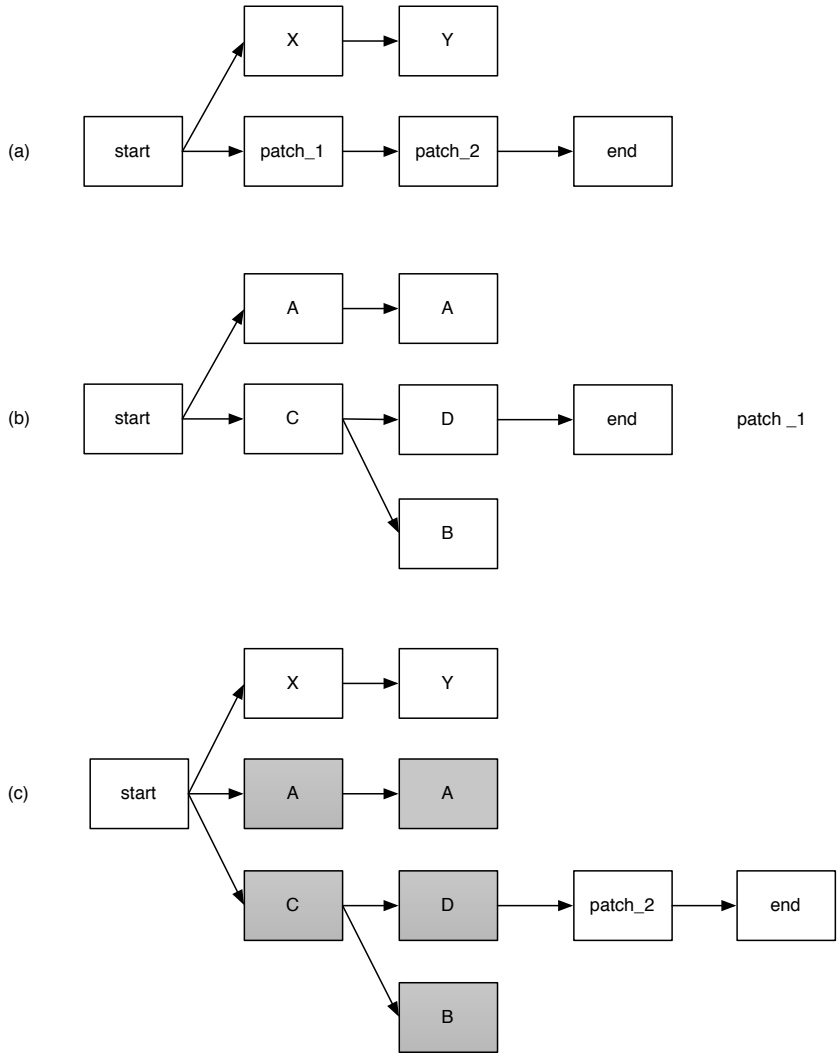


Figure 5.5: Illustration of the process of embedding a patch into another patch

5.5.1 User-defined groupings

In the first case, groupings can be supported by the interface as part of the creation activity, so that parts can be exchanged between participants. In this case, expansion rules are defined structurally by these groupings, and grammars can be inferred at different lexical levels. Here, frequent patterns can be identified as isomorphic trees. Two trees are considered isomorphic if there exists a mapping between the nodes and the edges of both trees that preserves the structure. In practice, it can be said that they are the same tree but can be drawn in different layouts. Isomorphic labelled trees can be found using algorithms such as VF2 (Cordella et al., 2004), which builds and explores a search space of partial mappings between two graphs. Also, for user-defined groupings it is possible to use edit distances to define similarities between sample patches. Tree edit distances (Bille, 2005) work in the same way as string edit distances, by defining a set of operations for converting one tree into another (node addition, node removal, and node relabeling). Ideally, different costs can be defined for these operations. In our case, since labels are generated by content-based audio analysis, relabeling cost can be defined by distances between content-based representations of clusters or classes, such as the distance between cluster centroids. The most classic algorithm for computing tree-edit distances was based on the simplification of assuming that the order between siblings in trees is meaningful (Zhang and Shasha, 1989). However, this limitation does not make sense for sample patches. More recently, the distance for unordered trees has been related to the problem of isomorphic subgraph matching (Bunke, 1997) which has allowed the definition of unordered tree edit distance algorithms (Torsello and Hancock, 2003).

5.5.2 Automatic groupings

Another use case for defining grammar rules is assuming larger trees generated by user activity, such as in real-time music performance sessions where users do not explicitly define groupings in the generation of the tree, although they may be implicitly repeating patterns. In this case, the identification of frequent patterns can be done through Frequent Subgraph Discovery (Kuramochi and Karypis, 2001) algorithms. Several algorithms have been defined that can deal with rooted trees: gSpan (Yan and Han, 2002), Sleuth (Zaki, 2005) or Subdue (Holder et al., 1994).

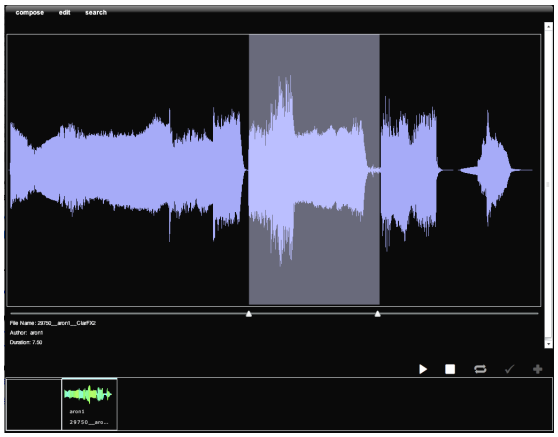
5.6 Experiments

Evaluation of music representations is complicated, especially by the fact that many are proposed on aesthetic grounds. Early experiments with grammars for music composition, and particularly the idea of mapping grammar alphabets to sound objects Roads (1978) were motivated by aesthetic factors. With respect to the framework proposed by Wiggins et al. (1993), the representation proposed in this chapter would achieve a level of *structural generality* comparable to symbolic notations, and the maximum level of *expressive completeness*, attributed to waveform representations. Ultimately, fully evaluating the potential of the possibilities for finding patterns enabled by the approach proposed in this chapter would require larges quantities of user-generated data. However, development and support of a large scale web application are out of the scope of this thesis. As a proof of concept, we implemented a basic prototype that allows the creation of musical works using the described representation on top of Freesound. The user experiment was done before the development of content-based indexing approaches defined in Chapters 3 and 4, and thus search was based on text queries. Analysis of user creations was done at a later stage using content-based indexing. The interface consisted of a flash application that connected to a python back-end. This interface was based on three panels that describe a creative workflow, and a tray shared by all panels to keep a palette of samples (Figure 5.6):

- Sample tray: By default, the tray contained a blank node that represents silence. The tray allowed duplicating any object and particularly silence objects of different durations could be created.

- Search panel: The search panel allowed retrieving samples and existing patches from the database. Sounds could be searched by tag, file name or user name, and a sound duration limit was specified (by default 10 seconds). Patches could be searched by file name or user name. Selected objects were dragged to the tray.

- Edit panel: The edit panel allowed the user to modify the start and end points of a sample, thus creating a new clip. This operation produced a new entry in the global alphabet of terminal nodes. Since the user might be interested in adding several instances of this terminal to the patch, the edit settings modified a master copy represented by the visual element in the tray.



(a)

(b)

(c)

Figure 5.6: Screenshots of the user interface of the prototype: a) edit panel b) search panel c) composition panel

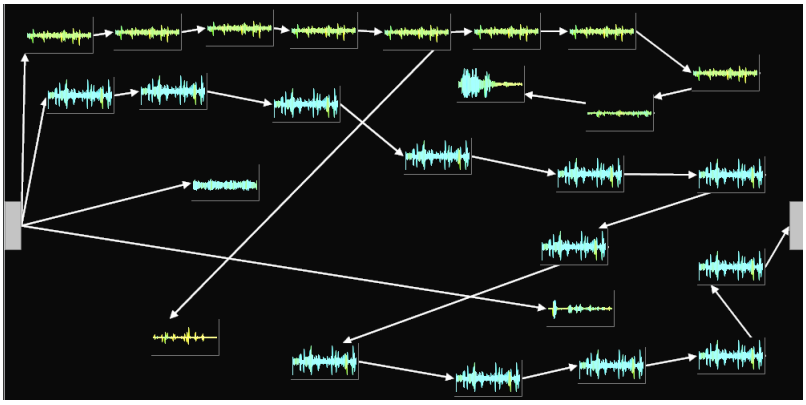
- Composition panel: In the composition panel the user could edit a sample patch with the samples and patches in the tray. Two rectangles in this panel represented the start and end terminal nodes. The user was then asked to create a composition where a path existed from the start to the end. When saving the patch the user decided whether to share it with the community or to keep it for herself. Users could continue to edit their own patches as long as they were not shared. When loading a patch to the tray, all the sounds and subpatches used in that patch were also loaded. Shared patches could no longer be modified (although new versions could be created through duplication). The reason was that modifying a shared patch could unexpectedly modify someone else's patch. Given more communication features, modification of shared patches could have been enabled in some cases for faster collaborative creation.

When the user saved a patch, the object structure was encoded in a JSON file, the patch was rendered to an audio waveform, and a thumbnail of the composition panel was generated. All files were sent and stored in the server.

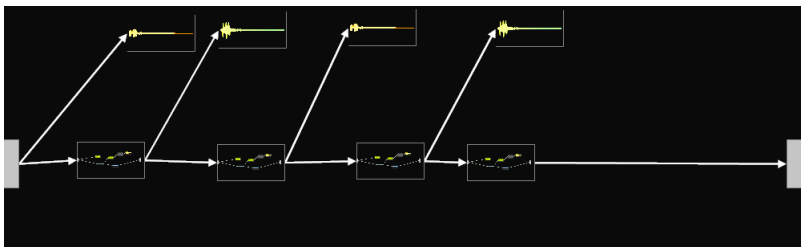
5.6.1 Case study

In order to test the idea of rooted tree grammars for undirected collaboration, we put the prototype online with some baisc usage explanations. During the initial trial period, we collected about 65 patches from 15 users. Of these users, most generated one or two patches, and two went on to create 16 and 21 patches respectively. We will call these users "A" and "B". The first one recognized himself as an expert using music production tools, but didn't have any programming or computer science background, while user B is a music technology graduate student. During this informal test, it was clear that people understood and used the possibility of nesting compositions at different levels. However we also found that more intensive use of this feature would require a pre-existing motivation for collaborating, for example an already established team. For example, Figure 5.7 shows a patch created by user A, who reportedly forgot about the possibility of nesting patches. Editing this kind of patch quickly becomes tedious, although adding some features to the interface could help. As a comparison, the patch below exploited this feature conveniently, which allows concentrating on the higher level structure and, in the case of individual use, the modification of the repeated portions of the piece at once. It became clear that

this may require some practice. With the data collected during the test, 69% of patches contained no nested structures, 18% contained one level and the remaining 8% more than one level. Almost half of the patches with a syntactic level higher than zero were generated by user B. On the other hand, some users nested other patches in their first creations. In all, 54% of all patches participated in some nesting relation, either contained or as containers. These tend to have a lower number of nodes (mean 6.9,sd 4.8) than patches with no nesting relation (mean 10.5,sd 7.6).



(a)



(b)

Figure 5.7: Examples of a) using / b) not using nested structures

5.6.2 Data analysis

In order to test the ideas about pattern detection based on sample patches explained in this chapter, we labelled the sample patches by applying the *Louvain* modularity clustering algorithm described in chapter 4. In total, 356 samples were used, which represents a very small fraction of the Freesound database. Also, the experiment was previous to our work on the

identification of different kinds of sounds reviewed in chapter 3, and the prototype did not focus on any specific type of sound. In order to adapt to this situation, we restricted the clustering to the sounds that were used in the experiment. The same ideas could be applied at a larger scale to applications based, for example, on music loops. The clustering done with mean and variance of 13 MFCC coefficients available at the time from the Freesound database. Using a value of 5 neighbors for k resulted in a taxonomy of 2 levels with 8 and 19 clusters respectively. For the sake of simplicity, we use the first level in our diagrams. Table 5.1 shows the size and the three most representative tags from Freesound for each of the clusters. Label 0 is reserved for start and end nodes, and 1 for silence, while -1 means that a sound had been deleted (by its author) from the database.

Table 5.1: Clusters from sounds used in the experiment

Label	N sounds	Tags (occurrences)
2	50	voice (7), horror (6), scream (6)
3	10	talk (7), vocal (7), female (7)
4	47	synth (9), pad (7), clarinet (6)
5	33	barcelona (10), boqueria (5), market (5)
6	53	loop (10), kick (7), drum (6)
7	49	hit (7), processed (6), noise (5)
8	38	echo (4), loop (4), noise (4)
9	41	percussion (7), loop (5), effect (5)
10	15	loop (7), street (3), ambient (3)

User-defined groupings

As we have seen, the restrictions imposed by the representation imply that sample patches can be directly interpreted as the rules of a grammar, at different levels defined by the lexical mapping. Figure 5.8 shows several nested patterns labelled at the first lexical level defined by the modularity clustering. The nested structures reflect the unorganized collaboration of three different users, authors of the 5 patches, as well as the authors of all sounds involved. In this case, different music pieces could be generated by choosing different sounds from the same clusters. On the other hand, characterization of users and groups could be done based on frequent isomorphic patches, which could also be automatically exchanged. We checked the pos-

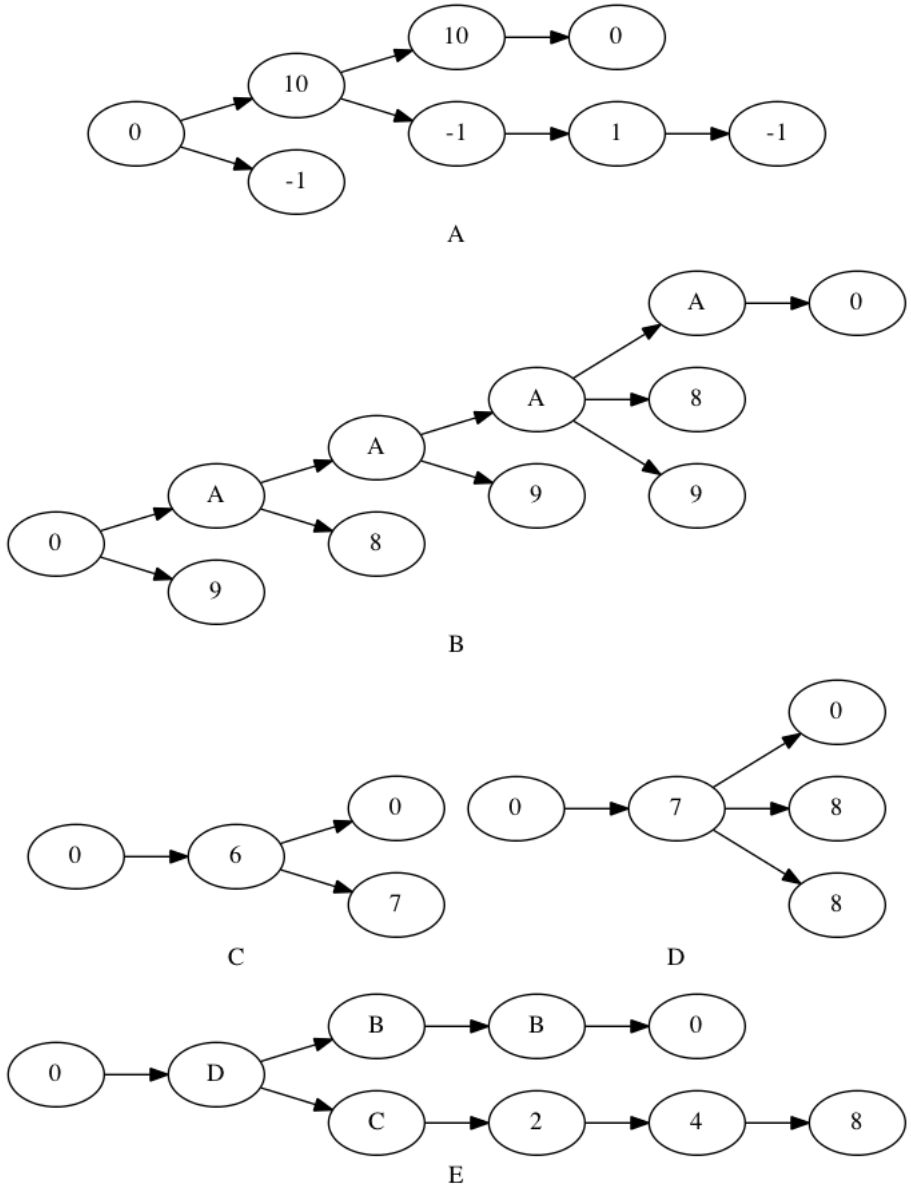


Figure 5.8: Rules defined by the patches of 3 different users

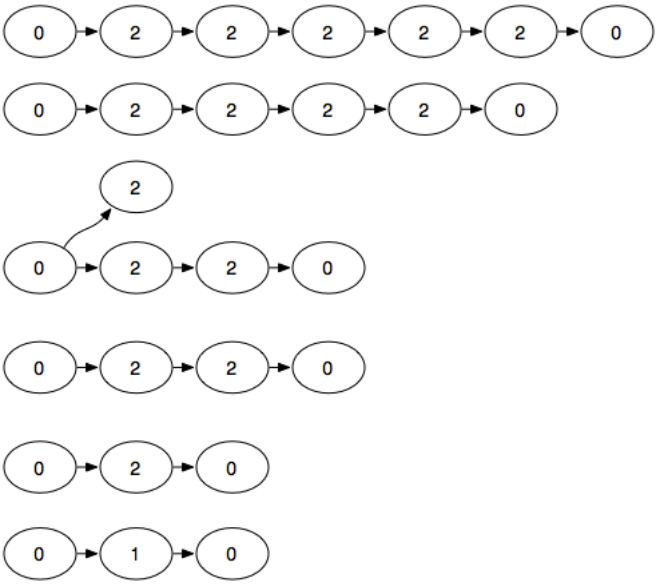


Figure 5.9: Repeated patterns found using VF2

sibility for finding isomophic patches using the VF2 algorithm to compare all terminal patches (this is, without nested subpatches) between them. In this case, there were only some basic repeated isomorphic patches the top lexical level (figure 5.9). Clearly, finding isomorphic patches would require a much larger quantity of user-generated data. In this sense, the experiment serves as a proof of concept. On the other hand, it is also possible to find automatic groupings as frequent subgraphs inside user-generated patches. We explored this possibility in the next experiment.

Automatic groupings

In order to analyze the possibility of finding sub-patch level patterns, we ran the Subdue program (Holder et al., 1994) on a file with all the patches labelled according to different lexical levels. Subdue accepts a *minsize* option to specify the minimum number of nodes of the identified subgraph. Obviously, the larger this number, the harder it will be to find frequent patches. By using several lexical generality levels (the top level that labels all sounds with the same value, two clustering levels, and the lowest level where each sound is labelled with its own node id) we could test the intuition that this number makes it possible to adjust the number of pat-

terns that are identified. Figure 5.10 shows the results for different *minsize* values on a logarithmic scale. Clearly, the number of detected subgraphs decreases monotonically for all sizes with the lexical level. This implies that this technique can be used to adjust for detecting patterns at different levels, potentially controlling for the number of sounds and users involved. For example characterizing individual users could be done by finding frequent patterns at lower levels, while large user groups would require higher levels. Figure 5.11 shows the most frequent subgraphs identified by Subdue with *minsize=4*. Considering that clusters 4 and 9 contain typically loops and electronic sounds (table 5.1), it seems clear that at larger scales this kind of analysis would allow to characterize the music style and composition strategies of users, in this case predominantly repetitive electronic music.

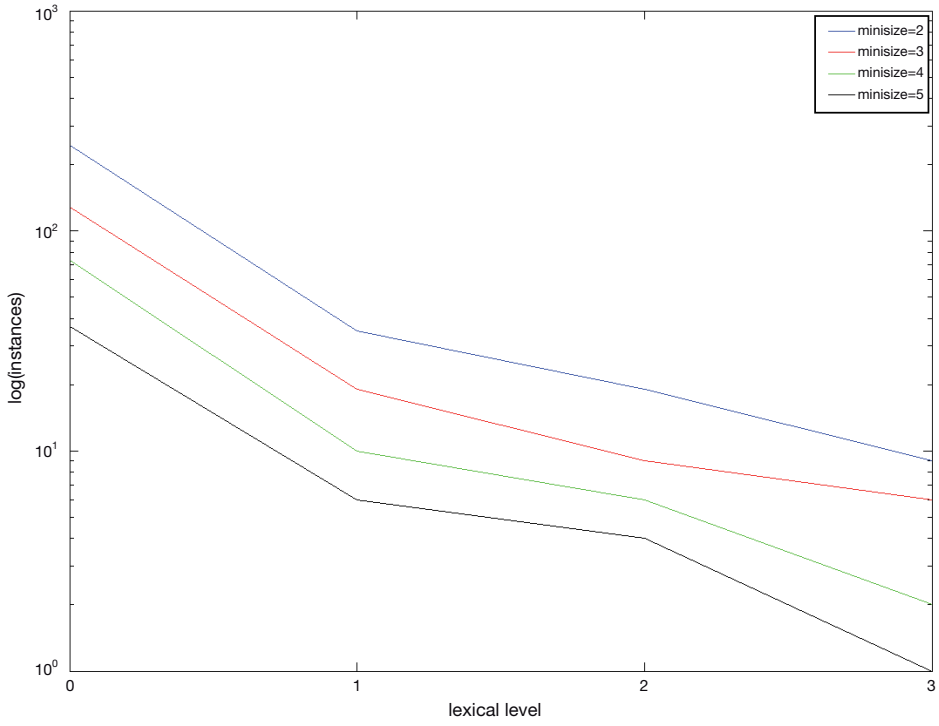


Figure 5.10: Number of identified subgraphs as a function of the lexical generality level for different graph sizes

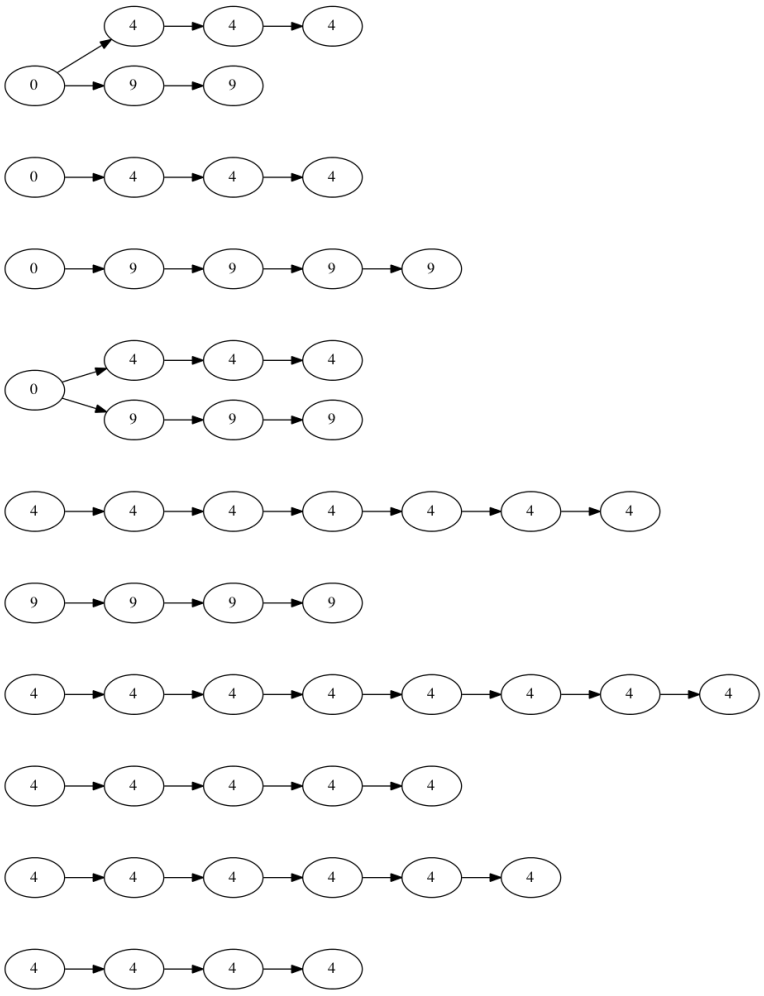


Figure 5.11: Frequent subgraphs identified by Subdue with minsize=4

5.7 Conclusions

The idea of music creation based solely on audio recordings should not be surprising for anyone familiar with electronic music tools. Considering a nascent medium with important limitations such as web audio, there are some specific affordances that can be exploited. One is taking advantage of large audio databases in traditional web development architecture models. Another is enabling collaboration. In this chapter we have presented a representation that can be used in web-based environments to create concatenations of audio samples. We proposed "sample patches", defined as rooted trees where nodes represent audio samples, as horizontal representations that are easy to understand and can be manipulated by computer programs. By defining an embedding mechanism for sample patches, it is possible to specify grammars that organize colaborative music making. While we have not analyzed specific workflows, it is clear that this representation would enable different musical practices with different levels of agreement between internet users. We conducted a proof-of-concept experiment that shows how this idea can be used to represent collaborative work. At the same time, it was clear that further investigation of specific workflows would be necessary in order to create engaging experiences. With data generated by users, we could also validate the idea of finding patterns in rooted trees using available implementations of graph mining algorithms. The notion of lexical generality level can be used for adjusting the generality of identified patterns using the approaches for taxonomical organization described in chapter 3. We expect that representations such as the one presented in this chapter can play an important role in collaborative applications by allowing them to take advantage of information sharing.

Chapter 6

Understanding networked creativity

6.1 Introduction

Social networks have become a very prominent usage of the web. Many flavors of social networking applications are frequently used by millions of users, many of them related to professional or hobby activities. It can be expected that, in a near future, user networks will play an important role in any activity based on computers, including music production. Social networks, and social phenomena in general are, however, complex and hard to understand. This makes it difficult to foresee the consequences of a particular feature or design in an application driven by social interaction. In addition, different users may have different goals and requirements, which challenges the definition of a general guiding principle for such applications. In this chapter, we experiment with the concept of collective creativity as a tool for understanding and devising applications for collective music making based on shared data. As reviewed in chapter 2, several authors have worked on the hypothesis that collective creativity is influenced by network topologies, particularly small-world network structures.

At the time of this writing, social networks for music creation are still in early development. Data on music creation networks (i.e. where the network facilitates the creation process) is generally not available. However, some analysis can already be done on communities for sharing audio clips that have been running for several years. Some already mentioned examples are Freesound or Looperman. Sharing audio can be seen as an initial step

towards audio-based collaborative music production. Audio clip sharing websites offer an interesting example of a collective process which produces a value that goes beyond individuals. The motivations that lead people to record and upload sounds for free are clearly related to social factors. The notion of creativity, as a concept that involves both *innovation* and *relevance*, can serve as a measure of the success of this process. We hypothesize that the analysis of several implicit user networks that can be extracted from audio sharing sites can give an insight on how this collective creative process works, and test these ideas analyzing data from Freesound. Most of the information used in our experiments is generally accessible through the website and could be extracted automatically from the web pages or, in many cases from the web API available for developers. For a user, this includes the sounds she has downloaded, the tags used for her own sounds, comments and ratings assigned to sounds, and the posts in Forums. In Freesound, all this information is publicly shared, which can be related to the open philosophy of the site, especially with respect to content licensing. For the case of music creation, it may be possible that users are interested in greater privacy, at least in some phases of the creation process. In any case, the analyses described in this chapter are based on statistics of anonymized data, and can be applied without compromising user privacy.

Most of the chapter was published in (Roma and Herrera, 2010a) and (Roma et al., 2012a).

6.2 Implicit networks in audio sharing

The three main entities that are stored in repositories of audio clips are audio files, users, and textual descriptions of audio files. Most sites use free tags as textual descriptions of the audio content. This structure is very similar, although not equal, to general tagging systems, which are traditionally studied as tripartite hypergraphs (Mika, 2005). In tagging systems, an annotation is seen as a link between a user, a resource and a label. From this representation, several networks can be extracted, for example between users on the basis of their shared resources or on the basis of shared tags. Audio clip sharing sites are different in that users normally describe the files they create and not those of other users. For example in Freesound, tags are mainly generated as part of the file uploading process and, while it is possible for any user to tag sounds, 99% of the tags are assigned by the author of a sound. A possible explanation may be that, while in tagging sites such as delicio.us or music radios like last.fm users act as active receivers

of information and their main activity is consuming and describing that information, in audio clip sharing there is a more practical motivation in using these files (i.e, the creation of music or audiovisual content) and less in directly enjoying them online. Following the idea that the community motivates and influences users in their contributions, we hypothesize that implicit user networks contain valuable information in order to understand the dynamics of audio clip file sharing.

6.2.1 Download networks

We define the download network of an audio clip sharing system as a graph where each node is a user and each edge represents a file that has been downloaded between two users. A user may download many files from the same other user, and thus the network is a multigraph. This network can be seen as an analogy to directed citation networks (White et al., 2004). However, for consistency with the rest of networks and with most of the small-world literature, we convert this graph to an undirected one. In this case, downloads between two users in both directions become a number of undirected edges that represent the strength of the similarity between both users.

6.2.2 Semantic networks

The *semantic network* of users of an audio clip sharing site can be derived from the concepts they employ to describe their sounds. In this case, nodes are also users, but edges represent the similarity between two users based on their tags. Thus, if we represent a user with the vector $v_i = v_{i0}...v_{in}$ where v_{ij} is the number of files that the user i has annotated with tag j, the adjacency matrix can be defined using cosine similarity:

$$a_{i,j} = \frac{v_i \cdot v_j}{||v_i||||v_j||} \tag{6.1}$$

This network can also be seen as a multigraph by considering a minimum weight unit (Newman, 2004) so that similarity is expressed in integer values.

6.2.3 Shared interest networks

Another kind of network can be extracted by counting the number of sounds that two users have downloaded in common. This type of co-occurrence tie can be thought of as an analogy to tagging systems (Mika, 2005). A

user is represented by a sparse vector that represents the sounds she has downloaded. We define the *shared interest* network by computing cosine distances between these vectors.

6.2.4 Forum and communication networks

Yet another way to represent the community can be made by measuring the amount of communication between users, for example analyzing the activity in a Forum. Forums are a common feature of online community sites for sharing content. We follow standard practice by linking users that have participated in the same threads. Hence again cosine similarity can be computed using the vector of threads.

6.3 Network analysis

Several properties can be used to characterize the described networks. We now describe some of the properties that can be related to creativity measures.

Density

The density of a network measures the number of links with respect to the amount of possible links given by the number of nodes. Thus it is simply defined as

$$D = \frac{nm_d}{n(n-1)} \tag{6.2}$$

where m_d is the mean degree and n the number of nodes. Thus, density reflects the level of activity in networks that represent or involve interactions, regardless of the number of nodes.

Assortativity

Assortative mixing refers to the tendency of nodes to connect to similar nodes, which can be measured by different criteria (Newman, 2003). One common measure is degree correlation, this is, the Pearson correlation of node degrees. In weighted networks, another possible measure is strength correlation, where strength is the sum of the weights of the edges connected to a node.

Efficiency

The traditional small-world network model defines that the average shortest path length is similar to the random graph. The average shortest path length measures the minimum number of edges that join any two nodes in the network,

$$SPL = \frac{1}{n(n-1)} \sum_{i,j} d(v_i, v_j) \tag{6.3}$$

where $d(v_i, v_j)$ is the length of the shortest path between nodes v_i and v_j. The problem with this property is that when the network includes disconnected components, the distance between disconnected nodes is theoretically infinite. A solution is to compute Efficiency, defined as the average of the inverse of the shortest path lengths:

$$Eff = \frac{1}{n(n-1)} \sum_{i,j} \frac{1}{d(v_i, v_j)} \tag{6.4}$$

When no path exists between two nodes, the efficiency is simply zero. Efficiency obviously grows with the number of edges of a network. In order to determine the significance of a certain efficiency value, it must be compared to an equivalent random graph. This graph is usually built according to the Erdős-Renyi (ER) model, which randomly wires a specified number of nodes given a probability p. Following the standard practice (Humphries and Gurney, 2008) with the average shortest path length, we compute the ratio of efficiency with that of the random graph:

$$Eff_r = \frac{Eff}{Eff_{rand}} \tag{6.5}$$

Clustering coefficient

In the original model by Watts and Strogatz, the clustering coefficient was computed as the average fraction of neighbors of a vertex that are connected between them. This is now usually known as the *local* clustering coefficient. For undirected graphs this is defined as

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{6.6}$$

where k_i is the degree of node i, and E_i counts the number of edges between the nodes connected to node i. The clustering of the network is then com-

puted as the average of the local clustering for all nodes. Like in the case of efficiency, we compute the ratio CC_r with the equivalent ER graph.

Small-world coefficient

Clustering and average path length are usually combined to give a measure of "small-worldness", defined as the quotient of $\frac{CC}{SPL}$. Here we follow the convention but using efficiency, so we define the small-world coefficient as

$$SWC = CC_r Eff_r \tag{6.7}$$

Modularity

We have already described modularity for clustering networks of similar sounds in chapter 4. This measure was originally developed for partitioning social and complex networks. Here, we use modularity clustering for finding groups of users, and modularity value obtained for the best partition as a measure of how divided is the network. For the case of user networks, we use modularity as defined for undirected multigraphs, which means that we take into account the strength of the edges as an integer representing the number of connections (e.g. downloads) between two nodes. Given the adjacency matrix A of a graph where A_{ij} is the number of links between nodes i and j, modularity is defined as

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(v_i v_j) \tag{6.8}$$

where m is the total number of edges in the network, and $\delta(v_i v_j)$ is a function that returns 1 if the group of node i and node j are the same and 0 otherwise. Values above 0.3 are usually considered to indicate a modular structure (Newman and Girvan, 2004)

Given this definition, many algorithms have been described for partitioning a network by directly optimizing modularity. We use the Louvain method (Blondel et al., 2008) described in chapter 4.

6.3.1 Static vs dynamic analysis

All of the described measures can be extracted from networks that are studied as static entities. Complex networks analyzed in many disciplines can be seen as a snapshot of a given state of a complex system, or as the final state

of some process. Analysis of the evolution of the network through different states is also common (Carley, 2003). In the case of audio clip sharing sites, the network can change significantly over time as new users register and upload more files, while other users stop visiting the site. In order to capture the interactions of active users, in addition to traditional static analysis we consider a dynamic approach where we extract the networks using only information about events happening in a given time period. In the case of the downloads network, this sample will include users that have uploaded sounds but not logged in that time frame. For the rest of networks, only active users appear as nodes in the network. For the purposes of this study, initial experiments with "accumulated" networks (where all past events are considered at each sample) yielded similar results at much higher computational cost, but the extracted measures suffered from high multicollinearity, which made them unfeasible for regression analysis. Hence, only interactions happening within one sample period are considered.

6.4 Creativity measures

The concept of creativity can be helpful in understanding online behavior in online audio clip sharing. This notion is commonly considered to be composed of a *novelty* aspect and a *relevance* aspect. These properties can be attributed to creative artifacts in the context of a given community. One sound can be more novel and/or relevant for the community regardless of who has uploaded it, and the same individuals can be more or less creative depending on the moment and the community who judges their creations. For accounting the novelty component, in the next section we consider an automatic approach that objectively describes the originality of a given sound. Since no user is assumed to have listened to all of the sounds in the database, it would be difficult for anyone to judge the originality of a sound for the whole community. Whether a sound is relevant or not seems a much more difficult question for a computer so we consider feedback measures provided by the community.

Novelty

The issue of determining the novelty of a document has been considered in text retrieval (Tsai, 2010). A simple approach to document level assessment of novelty is to simply consider the distance to the closest neighbor of a document. In the context of audio clips, the distance can be computed using content-based descriptors as described in previous chapters. This

allows us to compute the originality of a sound in a dataset as the distance to its closest neighbor. In order to obtain a novelty value for a given time frame, we use a KD-tree data structure (Bentley, 1975), where we add all sounds of the first time frame. For each subsequent time frame, we average the distance of each sound introduced in that time frame with the closest one in the tree. This average gives us an estimate of the amount of novel information introduced in the database for that period. After all sounds in a given time frame have been processed, they are added to the tree ad the next time frame is processed.

Relevance

A common feature in web applications is to allow users to participate at least in the form of comments and ratings. Depending on the sites, some of these functionalities may be available. On the other hand, an objective measure of the success of a sound is the number of times it has been downloaded. In the case of comments and ratings, the situation may vary. In the case of Freesound.org, comments and ratings are sparse and always very positive, so their mere presence is always an indicator of value of a sound. Thus, in this case we can use three measures: number of downloads, number of comments and number of ratings as indicators of the relevance of a clip to the community. Since the number of people who download, comment or rate sounds is much larger (by a factor of 500) than that of the users who upload sounds, we can consider these measures as measures of the success of the community of uploaders among their audience. The value for a given time frame is obtained in the following way: for the sounds that are added in that time frame, we consider the number of downloads, comments and ratings that the sound gets during its life since then until the last date of the analysis. Since the probability of getting feedback and downloads increases with the age of a sound, we normalize these values by the number of days that sound have been in the database. We then average for all sounds in the time frame to obtain an indicator of the creative performance of the community for that month.

6.5 Experiments

6.5.1 Community structure in Freesound

In recent years, networks have been increasingly studied as a means for understanding complex phenomena such as social groups and processes.

Several characteristics have been observed in real world networks as opposed to basic mathematical models such as regular lattices or random graphs. Particularly, many real world networks can be characterized as small worlds, and it is also common to find modular structures. Groupings in these networks are usually referred as "communities" in the complex network litterature. Thus, from this point we will have to use this concept in a more technical way, signifying a group of nodes in a user network that have more connections between them than they have with the rest of the network. An early experiment with the Downloads and Semantic networks extracted from Freesound revealed that the social structure of the site reproduces these common findings, and provided some hints about the kind of social organization that can emerge from audio sharing tools. In order to characterize the user communities identified by network analysis, we used the tags employed by users to describe their sounds. As expected, the semantic network, built using those tags, shows more clearly defined themes in this respect, but it is also interesting to see whether the communities of the download network are related with different semantic topics. We use the probability of a tag in a community $P_{i|C_k} = \frac{n_{i|C_k}}{n_{C_k}}$ (the number of documents annotated by tag i in community k divided by the total number of documents in that community) and the analogue for the whole network $P_i = \frac{n_i}{n}$. The ratio P_C/P_G indicates if a tag is commonly used in the community with respect to its normal use in the whole site. Values above 1 indicate a more specific use in this community, and tags with large values are used to characterize the community. On the other hand, the entropy of a community with respect to tags can be described using the local probabilities:

$$S_{C_k} = -\sum_{i \in C_k} P_{i|C_k} log P_{i|C_k}, \tag{6.9}$$

in the same sense as the genre entropy used in studies of music communities (Jacobson et al., 2008). An average of this quantity over all communities (S_C) provides a hint about the quality of the partition with respect to tags. A low value of the entropy of tags in the community indicates a greater homogeneity with respect to tags. Note that, usually, tags are not necessarily as general and comprehensive as genres are in music. However, for this measures we limited tags to the most popular ones, with a frequency of more than 1000 in the database. On the other hand, since the value of entropy depends on the size of the community, we computed the entropy of a proportional random partition of the network as a baseline.

Download network

One thing that stands out when building the download network of Freesound, is that a vast majority of users are only interested in downloading files. The reason is that, in order to download files at their original quality, the site requires people to sign in. These users could be represented by nodes with zero in-degree. However, the purpose of this study was to characterize the network of users who upload sounds, and we discarded users who did not upload sounds. This gives us a reduced network of 2234 nodes (out of the more than a million of registered users) and 60342 edges. In this reduced network, we find a strongly connected component of 1836 nodes, about 80% of people who have uploaded some sound. The remaining 20% is mainly split among users who have zero in-degree (this is, people whose sounds have not been downloaded by any other author) or zero out-degree (users who have uploaded sounds but that have not downloaded any sound), with a remainder of nodes that mostly have either one incoming or one outgoing link. This structure mimics the *bow tie* structure that has been found to characterize the World Wide Web (Broder et al., 2000)(Table 6.1).

Table 6.1: Global structure of the download network: Strongly Connected Component (SCC), nodes with zero in-degree (IN) and nodes with zero out-degree (OUT)

Part	N of nodes	Percent	Average in-degree	Average out-degree
SCC	1836	82%	32.33	31.68
IN	145	6.5%	0	21.04
OUT	187	8.4%	11.24	0
Other	66	2.9%	10.47	14.32

On the other hand, the network of authors can be characterized as a small world. Table 6.2 shows the average shortest path length and clustering coefficient of the network compared to the equivalent ER graph. While the average distance between nodes is roughly the same than for the random graph, the clustering coefficient is an order of magnitude larger. Interestingly, assortativity is negative, which means that nodes tend to associate with nodes of different degree.

The application of the Louvain method for community detection to this net-

Table 6.2: Main properties of the download network (in parenthesis, values for the equivalent ER graph)

Download network properties	Values
Nodes	2234
Edges	61697
Mean degree	27.62
Diameter	5
Assortativity	-0.181
SPL	2.23 (2.22)
Clustering Coefficient	0.28 (0.024)

work reveals a structure dominated by a large community with a modularity of 0.29. Table 6.3 shows the different communities and their characteristic tags. While this structure may seem similar to the global structure described by the component analysis, as a matter of fact the proportion of nodes of each community that belong to the strongly connected component is very similar (around 80% for all communities). This discards that the modularity partition merely replicates the division between the giant component and the rest. At the second level, this large component loses some nodes to small sibling communities, but the trend is preserved. An analysis of the most characteristic tags in the top level (Table 6.4) shows, however, that there are some differences in the topics preferred by each community. The main community, while more heterogeneous in the tags, is characterized by one of the most popular tags of the site: *field recording*, and generally concepts related to environmental sounds. Contrastingly, in the smaller communities, concepts related with musical uses of sounds are more salient. However, the general entropy of the partition is not much lower than the entropy of the equivalent random partition.

Semantic network

The semantic network is itself an undirected, weakly connected component, although it contains less nodes (2161) than the download network. The missing 27 users are disconnected from this network because they use few and very specific tags that nobody else uses. On the other hand, the number of edges is higher (75607) which makes a very dense network.

An analysis of the average shortest path length and the clustering coefficient

Table 6.3: Number of communities, modularity, entropy and average entropy of equivalent random partition in download network

Download network tags	Values
Communities	5
Modularity	0.29
S_C	3.02
S_C (random)	3.17

shows that this network can also be characterized as a small world (Table 6.5). This feature is even more pronounced than in the download network. Again, assortativity is strongly negative, which is understandable from the construction of the network: users who use many words to describe their sounds will be linked to users who use less (but similar) words and therefore have a smaller degree.

Modularity-based community detection reveals a more pronounced and also more balanced division of the network with respect to the download network, with a modularity of 0.35 (Table 6.6). Here, the distribution of topics is clearer among the different communities (Table 6.7), as could be expected from the construction of the network (for example, *techno* appears in a community with related topics, while in the download network it appeared in the larger community along with concepts related to environmental sounds). Also, the average entropy is significantly lower than the value for the equivalent random partition. Several groups are related to percussive sounds used for creating rhythms in electronic music, while others seem more related with voice (another popular tag of the site) or environmental sounds. The case of voice samples is characteristic: a tradition exists in Freesound to provide recordings of voices with specific utterances upon request. This community could be considered to reflect this activity. Some of the detected communities are very small and do not contain any of the more popular tags, so they are omitted from Table 6.7.

Confusion of communities in both networks

Table 6.8 shows the amount of users in each community of both the semantic and download networks. Row and column indexes correspond to indexes of the communities in each network. Both partitions seem to be related, with the main group of the download network splitting equally into two of the

Table 6.4: Characteristic tags in communities of download network (probability ratio)

Download network communities	Probability ratio	Characteristic tags
Community 0, Size: 33, Entropy: 2.58	5.49	human
	8.67	acoustic
	2.92	short
	2.01	percussion
	1.55	field-recording
	1.28	noise
Community 1, Size: 294, Entropy: 3.26	5.26	analog
	2.52	drums
	2.10	percussion
	1.91	drum
	1.80	beat
	1.80	glitch
Community 2, Size: 32, Entropy: 2.38	6.90	percussion
	6.18	metal
	5.45	drum
	4.66	hit
	2.45	guitar
	2.30	drums
Community 3, Size: 1644, Entropy: 3.48	1.59	birds
	1.59	nature
	1.53	male
	1.50	ambience
	1.42	field-recording
	1.37	techno
Community 4, Size: 231, Entropy: 3.41	2.84	reaktor
	2.76	multisample
	2.22	drone
	2.08	space
	1.61	fx
	1.54	electronic

Table 6.5: Main properties of the semantic network (values for the equivalent ER graph)

Semantic network properties	Values
Nodes	2161
Edges	75607
Mean degree	69.97
Diameter	5
Assortativity	-0.25
SPL	2.07 (2.07)
Clustering Coefficient	0.54 (0.032)

Table 6.6: Number of communities, modularity, entropy and average entropy of equivalent random partition in semantic network

Semantic network tags	Values
Communities	6
Modularity	0.36
S_C	1.94
S_C (random)	2.79

semantic communities. For the rest, a majority of users of a community in one network belongs to a community of the other. A χ^2 test on this matrix, as a contingency table with 20 degrees of freedom returns a very small value ($p = 9.5 \cdot 10^{-24}$) which gives support to the rejection of the null hypothesis stating that both assignments are independent.

Conclusions about network structure

This first study analyzed two implicit user networks with data accumulated over time in the Freesound database. The analysis allowed us to learn about the structures that emerge from the community based on the rules for sharing audio under CC licenses (note that we now switch back to the more general concept of community). Both networks can be characterized as small worlds with a modular structure, as commonly observed in real-world networks and related with complex social phenomena. The downloads network allowed us to have a general understanding of the structure of the

Table 6.7: Characteristic tags (probability ratio) in communities of semantic network (communities with less than 10 nodes are omitted)

Semantic network communities	Probability ratio	Characteristic tags
Community 0, Size: 757, Entropy: 3.04	8.00	metal
	7.34	water
	6.35	hit
	3.58	birds
	2.56	percussion
	2.38	ambience
Community 1, Size: 128, Entropy: 2.62	11.90	human
	7.91	male
	7.71	voice
	3.18	short
	2.29	acoustic
	1.90	effect
Community 3, Size: 332, Entropy: 2.5	5.53	nature
	5.40	field-recording
	4.62	birds
	2.97	male
	2.78	water
	1.79	voice
Community 5, Size: 990, Entropy: 3.46	1.32	techno
	1.32	multisample
	1.31	electro
	1.31	reaktor
	1.29	analog
	1.28	glitch

Table 6.8: Confusion matrix: count of nodes in the detected communities of the download (rows) and semantic (columns) networks

	0	**1**	**2**	**3**	**4**	**5**
0	5	3	0	7	0	17
1	61	12	1	36	2	171
2	5	2	0	1	0	24
3	640	98	6	268	3	601
4	34	11	0	19	0	162

site as a giant component with appended IN and OUT components. On the other hand, the semantic network allowed a better identification of groups of users interested in different topics, which intuitively can be expected from the wide variety of sounds in the site. Another interesting fact is that these networks tend to show negative assortativity, which reflects that users do not group according to their degree (which can be related with social status in social networks), but tend to interact or share concepts with users that are different in this respect.

6.5.2 Predicting creativity

Beyond structural analysis, we were interested in the relation of these structural aspects with the ability of the community to provide increasingly novel and good quality sounds. We computed all of the described implicit networks, as well as the described creativity measures, by sampling the database monthly over a period of 66 months, from April 2005 to October 2010. We then related network properties with creativity measures through regression analysis. Table 6.9 shows the mean and standard deviations of the different properties of these one-month networks averaged over time. All of the described networks exhibit similar characteristics. With a few exceptions, they have negative assortativity in both degree and strength. Also with one exception (the shared interest network) they have high modularity, above the empirical threshold of 0.3 commonly considered to indicate the presence of modular structure (Newman and Girvan, 2004). The clustering coefficient ratios show that the networks tend to be much more clustered than the equivalent random graph, while the efficiencies are similar to random graphs, and so the efficiency ratios revolve around 1. Hence, the small-world coefficients (the product of efficiency and clustering) generally follow the clustering coefficient. This reflects that the measured networks follow

the small-world network model. Since the evolution of small-world coefficient is strongly tied to the clustering coefficient, and it does not make sense to include it in the same models as its factors, we omitted this property from the regression analysis.

Table 6.9: Mean (standard deviation) of the described properties for the different networks: degree correlation (dc), strength correlation (sc), modularity (m), clustering coefficient ratio(cc), efficiency ratio (ef), small-world coefficient (swc) and density (den)

Variable	Downloads	Semantic	Shared	Forum
dc	-0.13 (0.05)	-0.17 (0.06)	0.33 (0.31)	-0.22 (0.15)
sc	-0.01 (0.00)	-0.01 (0.01)	-0.01 (0.02)	0.04 (0.10)
m	0.45 (0.06)	0.40 (0.05)	0.17 (0.16)	0.55 (0.16)
cc	5.17 (1.52)	3.33 (0.73)	7.31 (2.55)	4.87 (2.07)
ef	1.14 (0.05)	0.97 (0.02)	0.85 (0.08)	0.71 (0.16)
swc	5.91 (1.86)	3.22 (0.68)	6.19 (2.16)	3.66 (2.19)
den	0.01 (0.02)	0.21 (0.05)	0.09 (0.05)	0.10 (0.14)

Regression analysis

In order to test the influence of the small-world properties of all four networks on the creativity measures we performed ordinary least squares regression analysis with the described variables for each of the networks. Our aim is to understand to which extent the properties of each network can model the creative outcome of the community. In order to account for potential causal relationships, we introduce a lag of one time period between the predictors and the dependent variables. For each model, we checked the variance inflation factor (VIF) to ensure that there are no collinearity problems. All of the VIF values were below 10, which is the usually recommended threshold (Hair et al., 2010) with the single exception of modularity in the shared interest network which went up to 10.8.

Tables 6.10 - 6.13 show the main regression data for predicting either novelty, downloads, ratings or comments, using the 4 different networks we have studied (downloads, semantic, shared interest and forum). For each dependent variable, the coefficients of the predictors are listed. Sign indicates the direction of the relationship (direct or inverse) and high absolute values indicate that the factor is relevant to predict that target variable.

Statistical significance is indicated with the usual asterisk coding (see caption for details). The bottom cell presents the determination coefficient R^2 which indicates the predictive accuracy of each regression function.

Table 6.10: Regression analysis coefficients for the downloads network. Significance codes: * ($p < 0.5$), **($p < 0.1$), ***($p < 0.01$)

Variable	Novelty		Downloads		Ratings		Comments	
dc	-26.36	*	-0.08		-0.01		0.00	
sc	-62.72		8.09	**	0.29	**	0.05	**
m	-20.01		0.53	***	0.02	***	0.00	**
cc	0.41		0.002	***	0.00		0.00	**
ef	7.44		0.008		0.00		0.00	
den	450.05	***	5.86	***	0.15	***	0.03	***
$\boldsymbol{R^2}$	**0.85**		**0.41**		**0.25**		**0.19**	

Table 6.11: Regression analysis coefficients for the semantic network

Variable	Novelty		Downloads		Ratings	Comments
dc	-38.18	***	-0.24	*	0.00	0.00
sc	-731.71	***	2.92	**	0.15	0.02
m	20.82		0.20		0.01	0.00
cc	3.78	*	0.00		0.00	0.00
ef	-15.07		-1.48	**	-0.05	0.00
den	111.23	***	1.51	***	0.04	0.00
$\boldsymbol{R^2}$	**0.76**		**0.49**		**0.36**	**0.09**

Discussion

The results show similar patterns with respect to the accuracy of the models, as measured by R^2. Novelty can be predicted with better accuracy in most of the networks. This seems more understandable if one takes into account that novelty is calculated using information from the same time period (except for the introduced lag) than the predictors. The rest of variables count the numbers of downloads, ratings and comments that sounds have received over time, which is a rough indicator of their quality but can obviously be influenced by other factors. On the other hand, network density appears to

Table 6.12: Regression analysis coefficients for the shared interest network

Variable	Novelty		Downloads		Ratings	Comments
dc	1.50		-0.11		0.00	0.00
sc	-87.23		-1.91	*	-0.02	0.00
m	20.08	*	-0.24		-0.01	0.00
cc	-0.03		-0.01		0.00	0.00
ef	-5.11		0.18		0.01	0.00
den	114.30	**	-0.28		-0.01	0.00
R^2	**0.76**		**0.40**		**0.24**	**0.05**

Table 6.13: Regression analysis coefficients for the forum network

Variable	Novelty		Downloads		Ratings	Comments
dc	-21.45	***	-0.17	**	0.00	0.00
sc	32.11	**	0.04		0.00	0.00
m	-6.97		-0.20	**	-0.01	0.00
cc	-0.92	*	-0.01	*	0.00	0.00
ef	4.06		0.00		0.00	0.00
den	52.69	***	-0.19		-0.01	0.00
R^2	**0.60**		**0.36**		**0.27**	**0.05**

be a generally dominant factor, along with strength correlation as a negative factor.

Novelty

The downloads network shows the highest accuracy with respect to the novelty variable, explaining almost 85% of its variance. The most important factor in this model is network density. This indicates that a higher number of downloading interactions between users may have a positive effect on innovation (recall that the network is computed only for active users who upload sounds). However, in the case of novelty, there is no reason to think that such potential causality would not go in the opposite direction. On the other hand, degree correlation has a negative impact, which could indicate that interactions among different users (in terms of degree) are positively related to novelty. This tendency is reinforced in those of the other networks

that best predict novelty. In the semantic network model, the strength correlation appears also as a negative significant factor. Modularity and clustering coefficient appear as small positive contributions in the semantic and shared networks. On the contrary, the effect of clustering of forum posts seems to be negative. In summary, novelty of uploaded sounds seems to be positively correlated with high connectivity, in some cases with a small bias towards clustering, and disassortative mixing.

Relevance

With respect to the relevance measures, the semantic network seems to have the greatest predictive power, followed by networks based on downloads (downloads and shared interest). For example almost 50% of the variance in the average number of downloads of a sound can be explained by the semantic network properties. In all of the networks, strength correlation appears along with density as an important positive factor. Also noticeable are the significant correlations of modularity and clustering in the downloads network. In contrast, degree correlation has a significant negative contribution in the semantic and also in the forum networks. This seems to point towards connectivity among active users (who should have higher strength regardless of the degree) as a factor for the relevance measures. Particularly in the semantic network, strength reflects the level of agreement of a user with other users in the tags that they use for the sounds. This describes an important connection between the people who upload sounds, which will affect the findability of sounds in the database. In contrast, efficiency (i.e. the harmonic mean of the average shortest path length) in the semantic network has a negative effect, which could mean that connecting remote users, who could be interested in different topics, with respect to text descriptions, is not beneficial. In conclusion, relevance seems to be also very related to network density in general and slightly with local density. Moreover, the agreement between users when describing sounds, reflected by strength correlation and density of the semantic network, seems to be among the most important factors for predicting relevance.

Relation to the small-world hypotheses

While the studied networks have shown to follow the small-world network model, our results show some differences with respect to the hypotheses reviewed in chapter2. The referenced works all analyzed real world social networks, which may follow different rules than networks mediated by in-

formation technology. In real world social networks, connections between distant clusters seem more difficult to achieve and therefore more valuable for diffusing information. In contrast, in internet-based communities, finding different people is very usual. In these communities, the difficulty lies in finding people with similar interests. In the context of the Freesound community this seems a reasonable explanation, since the site poses no restrictions or biases with respect to the kinds of sounds that can be uploaded, which results in a melting pot of many different cultures. Given network density as a general factor for the creative outcome, it would seem that local density and clustering, as reflected by the clustering coefficient ratio and modularity, are to be preferred to short path lengths. Our study also points towards the importance of assortative mixing, which is not usually covered in the small-world creativity literature. An interesting pattern in the analyzed data is that, while degree correlation (i.e. assortative mixing among nodes of similar number of connections to different users) is negatively correlated to creativity measures, strength correlation (which better reflects the level of activity of users in valued networks) has a positive influence in some cases.

Conclusions about networked creativity

It is often difficult to define the goals of a community based system, where each user may have different motivations.This makes it difficult to design and develop software for community usage. For the case of audio clip sharing, we have proposed a set of indicators of the creative outcome of the active community (users who create, or record, and upload sounds) using data from the larger community of information consumers. We have hypothesized that these measures partly depend on the dynamics of the community, which can be measured using the different networks of interactions in a given time frame. Our results show that part of the variance of these measures can in fact be explained by network measures. The proposed method, as well as the results of our empirical study, can be used to improve the design and functionalities of audio clip sharing sites. In this sense, it seems that features that try to maximize the density of the different networks, i.e. by facilitating actions that connect users, would impact on the quality and novelty of sounds that are uploaded. Similarly, features that promote connectivity among most active users, which would reflect on strength correlation, and features that promote clustering of users with similar interests are to be preferred.

As future work, the relationship between assortative mixing and creativity

could be further explored using other measures of similarity between users (beyond degree and strength), such as similarities between the sounds they download. Another possibility is analyzing the outcomes of individuals or smaller communities and their relationship to their network positions and centralities.

With respect to music creation based on audio, the described experiments should be a useful starting point, in the sense that collaborative audio-based music making is in itself at least an exchange of audio clips. However, music creation can be considered, in general terms, as a more complex activity than recording sounds (this obviously depends on the kind of music and the kind of recording). It may be expected that more complex relationships can be observed by considering similarities in music strucutre. In this sense, music representations such as the one proposed in chapter 4 could be used for computing novelty in the sense of music structure, detecting common patterns and similarities between users, and analyzing music collaboration networks using the methodology described in this chapter.

6.6 Conclusions

In this chapter we have analyzed users, as the last major type of entity implied in the use case of online music making communities. These communities can be seen as networks of users that create music by sharing information. By applying network analysis techniques, it is possible to gain a better understanding of the structure and dynamic behavior of the community. While users in online applications have diverse goals, we have proposed the concept of social creativity as a potential collective goal, that can be seen as the ability of the community to innovate with respect to the information stored in the database. We have tested these ideas by using information from the Freesound database, by defining user networks that are implicit to the activity of sharing sounds. While we hope that these methods can be applied to music creation communities in the future, the popularity of Freesound allowed us to test our ideas with a large-scale dataset. Our preliminary experiment showed that the proposed implicit networks exhibit characteristics found in other real world networks. On the other hand common network analysis measures can be used to describe the general structure of the audio sharing activity. We then performed a large scale experiment analyzing the dynamics of the implicit networks of the Freesound community. We analyzed how the proposed measures for analyzing creativity (an automatic content-based measure of novelty and

mesures of relevance based on downloading users) depend on properties of the networks. Our results indicate that high connectivity, represented by graph density, and negative assortativity (the tendency of nodes to interact with nodes of different degree) are indeed positively correlated with the creativity measures. This result should be of interest for supporting creative communities in general, and points towards a different behavior of online small-world creative networks with respect to existing literature analyzing networks of companies or teams creating musicals.

Chapter 7

Conclusions

During the last few years, the Internet, and particularly the web, has become even more of a crucial part of people's lives. Especially mobile platforms, basically smartphones and tablets, have contributed to fulfill the idea that everyone *needs* access to the web. At the same time, perhaps fostered by these developments, the adoption of new capabilities in web browsers has accelerated. At the time of this writing, the first edition of the first Web Audio Conference is about to start. The Web Audio API has been incorporated into most browsers. While audio software has traditionally remained in computer desktops, the specific affordances of the web are based on collaboration and information sharing. Advances in browser technologies, including mobile browsers, will necessarily improve our ability to collaborate in computer-based music creation. In recent news, the owners of the popular ProTools DAW software are announcing a new version with support for cloud-based storage. The general reliance on cloud computing may signal a second coming of efforts for developing internet-connected music creation tools.

This thesis explores the idea of using web applications based on large-scale shared data stores, particularly containing audio recordings, for music creation. This idea is backed by the success of existing platforms for sharing audio using CC licenses, such as Freesound. In this context, sharing audio affords a great potential for facilitating the creative development of users with diverse backgrounds on the basis of a long tradition of music based on audio recordings.

We started with the aim of adapting information retrieval techniques to support the use case of audio-based music creation by online communities.

In order to fulfill this goal, we analyzed three main entities that intervene in our use case: sounds, music fragments or pieces, and users. Providing and evaluating suitable representations and algorithms for this purpose has proved to be a challenge, particularly because of the unstructured nature of user-driven processes. One important axis of this work has been dealing with two opposite poles: at one side, we are interested in diversity, as a major component of creativity and innovation. At the other end, uniformity and consistency allow us to learn and efficiently take advantage of large-scale databases. Another important aspect of this thesis has been the cross-pollination resulting from the need to confront the different subjects in the use case of collective music-making based on shared data. The use of networks for modeling user communities has been applied to the sounds produced by these communities. Clearly, the distribution of audio features is derived from a certain social and cultural structure of users that share a given resource, such as Freesound. We have found that network analysis, and especially the concept of modularity, are particularly useful for dealing with this tension between uniformity and diversity.

In face of the complexity of the task, the development of this work has not necessarily followed the order of the dissertation. The later has been structured to follow the logic that can be used to design future applications for fostering musical creativity on top of shared audio. First, we have designed a framework that allows compact representations of different kinds of sounds, in a way that is useful for content-based retrieval. We expect that different kind of applications can be developed on the basis of different kinds of sounds commonly contributed by online users such as sound scenes, sound events and music loops. At the same time, we have provided algorithms for identifying these general kinds of sounds. We have then analyzed the two main machine learning paradigms (supervised and unsupervised algorithms) for facilitating access to large-scale audio databases in form of sound taxonomies. Both approaches can be used depending on the application, but for the case of unsupervised processes, a bottom-up approach such as modularity clustering shows promise for music creation applications. We have then proposed a representation for audio-based music that is simple enough for browser-based applications. We have shown that the grammar-based approach affords collaborative music-making based solely on the established practice of audio remix. At the same time, we have shown that in combination with taxonomical sound classifications, this representation allows the identification of common patterns in users creations. Finally, we have analyzed the structure of online communities currently sharing audio

files. In order to deal with the struggle between agreement and diversity, we have proposed the use of automatic measures of creativity, derived from computational creativity research, as a guide for applications that aim at fostering online music creativity based on shared audio.

Addressing this challenge lead us to investigate both existing and novel approaches to indexing and retrieval of each of the entities involved in the addressed use case. We now summarize the novel contributions of this thesis, which can be useful for information retrieval research and development focusing on large audio databases and audio sharing platforms.

7.1 Summary of contributions

7.1.1 Evaluation of low level generic features

MFCC features are very widespread and are commonly used without question alongside other frequency domain representations inspired by auditory filters. Our experiments with 9 different datasets showed that a change in the filterbank model can lead to some improvements in audio classification, which may not be visible with smaller datasets. This result should be of interest to content-based audio retrieval in general beyond our target use case.

7.1.2 Novel methods for feature aggregation

We have also shown methods for identification and summarization of different types of audio signals commonly found in online sharing sites, mainly environmental sounds and loops. We have adapted Recurrence Quantification Analysis features from research on nonlinear time series analysis and shown that they allow improving in audio classification tasks when added to traditional feature statistics.

7.1.3 Modularity-based audio clustering

The concept of graph modularity is often used for finding communities in social networks. We have shown that it is also useful for clustering audio in community-driven databases, where nearest neighbor graphs can be used to adapt to the uneven densities of the feature space. Using the *Louvain* multilevel modularity optimization algorithm allows creating automatic taxonomies of audio files. Comparing this method to a traditional supervised

method for taxonomical classification has shown that users find it easier to learn automatic unsupervised taxonomies.

7.1.4 Formal representation of audio-based music fragments

We have proposed a novel representation for music fragments defined by sequences and superpositions of audio files. The proposed representation specifically affords nesting music fragments at different levels using a graph embedding mechanism. We have shown that this representation supports the identification of frequent patterns in user creations by labelling similar sounds with content-based approaches such as modularity-based clustering.

7.1.5 Evaluation of sharing communities by creative outcome

Finally, we have proposed a method for measuring the creative outcome of an audio sharing community, using ratings from a larger *audience* community and an automatic content-based novelty measure. We have shown that these measures correlate significantly with measures extracted from analysis of user activity networks, particularly graph density and disassortative mixing. These results should be of interest for the design of applications aimed at fostering creativity such as online music creation platforms.

Some of these contributions have been published in conference proceedings and peer-reviewed journals. A list of the author's publications is available in Appendix A.

7.2 Future directions

With respect to the algorithms and representations proposed in this thesis, many possibilities remain for exploration. In dealing with a real world scenario such as the sounds existing in Freesound, we have faced the complexity of environmental audio, and proposed a segmentation approach. In addition to temporal segmentation, current research in source separation (which can be seen as spectral segmentation) could be used to expand the creative potential of audio contributed by internet users for music creation. From an implementation perspective, we haven't dealt with the issues of expanding a large database like Freesound using the event detection scheme

we have proposed. At larger scales, indexing approaches used for music such as locality-sensitive hashing should be investigated.

Regarding automatic taxonomies, we have generally avoided the analysis of labels used for audio collections. We have dealt with the generality of labels from a purely content-based perspective, but clearly a semantic analysis informed by natural language processing could help developing systems that are easily understandable to users. Also, in the case of unsupervised taxonomical organization, there are two important aspects that remain to be explored: one is choosing appropriate exemplars for each class in the taxonomy. Another is automatically finding intuitive labels, which could be helped by existing labels in sounds, or using automatic classifiers. In general combining bottom-up with top-down approaches could help dealing with data generated by unorganized user activity. In the case of music representation, we have dealt with the most basic case for enabling collaboration based on audio files. We have not investigated a way to derive probabilistic grammars that can be used for computer-aided composition. Also we have not dealt with more complex mappings potentially including transformations of audio files. Finally with respect to user communities, this thesis has only scratched the surface. We are in general still very far in understanding social behavior, particularly with respect to creative activities such as music. We hope that this thesis has contributed to the general objective of understanding music as a social phenomenon. Our contributions for defining similarities and groupings of user creations, and their relationship with social interaction, could be useful for future research on the social dimension of music creation.

Bibliography

Adenot, P., Wilson, C., and Rogers, C. (2013). W3C Web Audio API. W3C Working Draft, W3C. http://www.w3.org/TR/webaudio/.

Alvaro, J. L. and Barros, B. (2012). Musicjson: A representation for the computer music cloud. In *Proceedings of the Sound and Music Computing Conference (SMC '12)*, Barcelona.

Andries, M., Engels, G., Habel, A., Hoffmann, B., Kreowski, H.-J., Kuske, S., Plump, D., Schürr, A., and Taentzer, G. (1999). Graph transformation for specification and programming. *Science of Computer Programming*, 34(1):1–54.

Attali, J. (1985). *Noise: The Political Economy of Music*. Manchester University Press, Manchester.

Aucouturier, J.-J., Defreville, B., and Pachet, F. (2007). 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(2):881.

Baffioni, C., Guerra, F., and Lalli, L. T. (1984). The theory of stochastic processes and dynamical systems as a basis for models of musical structures. In Baroni, M. and Callegari, L., editors, *Musical Grammars and Computer Analysis*, pages 317–324. Leo S. Olschki, Florence.

Barbosa, A. (2006). *Computer-Supported Cooperative Work for Music Applications*. PhD thesis, Universitat Pompeu Fabra, Barcelona.

Battier, M. (2007). What the GRM brought to music: from musique concrète to acousmatic music. *Organised Sound*, 12(3):189–202.

Bel, B. and Kippen, J. (1992). Modelling music with grammars: formal

language representation in the Bol Processor. In Marsden, A. and Pople, A., editors, *Computer Representations and Models in Music*, pages 207–238. Academic Press, London.

Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. B. (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047.

Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, New Haven, CT, USA.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific American*.

Beygelzimer, A., Kakade, S., and Langford, J. (2006). Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104, Pittsburgh.

Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Boden, M. (2003). *The Creative Mind: Myths and Mechanisms*. Routledge, London, 2nd edition.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2006). Maximizing modularity is hard. *arXiv preprint physics/0608255*.

Brazil, E., Fernstroem, M., Tzanetakis, G., and Cook, P. (2002). Enhancing sonic browsing using audio information retrieval. In *Proceedings of the International Conference on Auditory Display (ICAD-02)*, pages 132–135, Kyoto.

Bregman, A. S. (1994). *Auditory Scene Analysis: The Perceptual Organization of Sound*. A Bradford Book, Cambridge, MA.

Bridle, J. and Brown, M. (1974). An experimental automatic word recognition system. *JSRU Report*, 1003(5).

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. *Computer Networks*, 33(1-6):309–320.

Brossier, P. (2006). *Automatic Annotation of Musical Audio for Interactive Applications*. PhD thesis, Queen Mary, University of London, London.

Brown, G. J. and Cooke, M. (1994). Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336.

Bryan-Kinns, N. (2004). Daisyphone: the design and impact of a novel environment for remote group music improvisation. In *Proceedings of the 5th conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04)*, pages 135–144, Brisbane.

Bryan-Kinns, N. (2012). Mutual engagement in social music making. In *Intelligent Technologies for Interactive Entertainment*, pages 260–266. Springer.

Bryan-Kinns, N. and Healey, P. G. (2006). Decay in collaborative music making. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME '06)*, pages 114–117, Paris.

Bunke, H. (1997). On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694.

Burk, P. L. (2000). Jammin' on the web – a new client/server architecture for multi-user musical performance. In *Proceedings of the International Computer Music Conference (ICMC 2000)*, pages 117–120, Berlin.

Cai, R. (2005). Unsupervised content discovery in composite audio. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 628–637, Singapore.

Cano, P., Koppenberger, M., and Wack, N. (2005). An industrial-strength content-based music recommendation system. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 673, Salvador.

Carley, K. M. (2003). *Dynamic network analysis*. Committee on Human Factors, National Research Council, National Research Council.

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

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

Chapelle, O., Haffner, P., and Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064.

Chechik, G., Ie, E., Rehn, M., Bengio, S., and Lyon, D. (2008). Large-scale content-based audio retrieval from text queries. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval (MIR '08)*, page 105, Beijing.

Chion, M. (1983). *Guide des Objets Sonores: Pierre Schaffer et la Recherche Musicale*.

Chomsky, N. (1957). *Syntactic Structures*. Mouton and Co.

Chu, S., Narayanan, S., and Kuo, C.-C. J. (2009). Environmental sound recognition with time-frequency audio features. *IEEE Transactions on Speech, Audio, and Language Processing*, 17(6):1142–1158.

Chu, S., Narayanan, S., Kuo, C.-C. J., and Mataric, M. J. (2006). Where am I? Scene recognition for mobile robots using audio features. In *2006 IEEE International Conference on Multimedia and Expo*, pages 885–888.

Clavel, C., Ehrette, T., and Richard, G. (2005). Events detection for an audio-based surveillance system. In *IEEE International Conference on Multimedia and Expo (ICME 2005)*, pages 1306–1309.

Cope, D. (2004). *Virtual Music: Computer Synthesis of Musical Style*. Mit Press.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372.

Cowan, R. and Jonard, N. (2003). The dynamics of collective invention. *Journal of Economic Behavior & Organization*, 52(4):513–532.

Cox, C. and Warner, D., editors (2004). *Audio Culture: Readings in Modern Music*. Continuum International Publishing Group.

Csikszentmihalyi, M. (1999). Implications of a systems perspective for the study of creativity. In Sternberg, R., editor, *Handbook of Creativity*. Cambridge University Press.

Dannenberg, R. B. (1993). A brief survey of music representation issues, techniques, and systems.

Dargie, W. (2009). Adaptive audio-based context recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(4):715–725.

de A Costa, W., Assis, F. M., Neto, B. G. A., Costa, S. C., and Vieira, V. J. D. (2012). Pathological voice assessment by recurrence quantification analysis. In *Proceedings of Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*, pages 1–6.

Duckworth, W. (2005). *Virtual Music: How the Web Got Wired for Sound*. Routledge.

Dupont, S., Dubuisson, T., Urbain, J., Sebbe, R., d'Alessandro, N., and Frisson, C. (2009). AudioCycle: Browsing musical loop libraries. In *Seventh International Workshop on Content-based Multimedia Indexing (CBMI '09)*, volume 0, pages 73–80, Los Alamitos, CA, USA. IEEE Computer Society.

Ellis, D. P. and Lee, K. (2006). Accessing minimal-impact personal audio archives. *IEEE Multimedia*, 13(4):30–38.

Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. Technical report. http://www.ee.columbia.edu/ dpwe/resources/matlab/rastamat/.

Ellis, D. P. W. (2009). Gammatone-like spectrograms. Technical report. http://www.ee.columbia.edu/dpwe/resources/matlab/gammatonegram.

Engelfriet, J. and Rozenberg, G. (1997). Node replacement graph grammars. *European School on Graph Transformation*.

Eronen, A., Peltonen, V., Tuomi, J., Klapuri, A., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. (2006). Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329.

Ertoz, L., Steinbach, M., and Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, pages 105–115.

Fastl, H. and Zwicker, E. (2001). Psychoacoustics: facts and models.

Fleming, L., King, C., and Juda, A. I. (2007). Small worlds and regional innovation. *Organization Science*, 18(6):938–954.

Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty. In *2000 IEEE International Conference on Multimedia and Expo (ICME 2000)*, volume 1, pages 452–455.

Foote, J. and Uchihashi, U. (2001). The beat spectrum: a new approach to rhythm analysis. In *IEEE International Conference on Multimedia and Expo (ICME 2001)*, pages 881–884.

Freeman, J. (2008). Graph theory: linking online musical exploration to concert hall performance. *Leonardo*, 4(1).

Freeman, J., Ramakrishnan, S., Varnik, K., Neuhaus, M., Burk, P., and Birchfield, D. (2005). The architecture of auracle: a voice-controlled,

networked sound instrument. *Network*, 5(6):7.

Ganchev, T. (2005). Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the SPECOM*, pages 191–194.

Gaver, W. (1993). What in the world do we hear?: An ecological approach to auditory event perception. *Ecological Psychology*, 5(1):1–29.

Geiger, J. T., Schuller, B., and Rigoll, G. (2013). Recognising acoustic scenes with large-scale audio feature extraction and SVM. Technical report.

Giannoulis, D., Benetos, E., Stowell, D., Rossignol, M., Lagrange, M., and Plumbley, M. D. (2013). IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events. online web resource.

Gillet, O. and Richard, G. (2004). Automatic transcription of drum loops. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processings (ICASSP '04)*.

Gillet, O. and Richard, G. (2005). Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems*, 24(2-3):159–177.

Gómez, E. (2006). *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona.

Good, M. (2001). MusicXML for notation and analysis. In Hewlett, W. B. and Selfridge-Field, E., editors, *The Virtual Score: Representation, Retrieval, Restoration*, volume 12, pages 113–124. MIT Press, Cambridge, MA.

Grey, J. M. (1977). Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277.

Guastavino, C. (2007). Categorization of environmental sounds. *Canadian Journal of Experimental Psychology*, 61:54–63.

Guaus, E. (2009). *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. PhD thesis, Universitat Pompeu Fabra.

Guo, G. and Li, S. Z. (2003). Content-based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks*, 14(1):209–215.

Hair, J. F., Black, W. C., Babin, B. J., and Anderson, R. E. (2010). *Multivariate Data Analysis: A Global Perspective*. Pearson Education, London, 7th edition.

Haro, M. (2008). Detecting and describing percussive events in polyphonic

music. Master's thesis, Universitat Pompeu Fabra.

Hermann, T., Hunt, A., and Neuhoff, J. G. (2011). *The Sonification Handbook*. Logos Verlag Berlin.

Herrera, P., Dehamel, A., and Gouyon, F. (2003). Automatic labeling of unpitched percussion sounds. In *Audio Engineering Society Convention 114*.

Herrera, P., Yeterian, A., and Gouyon, F. (2002). Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques. In *Music and Artificial Intelligence*, pages 69–80. Springer Berlin Heidelberg.

Herrera-Boyer, P., Klapuri, A., and Davy, M. (2006). Automatic classification of pitched musical instrument sounds. In *Signal Processing Methods for Music Transcription*, pages 163–200. Springer.

Hewlett, W. and Selfridge-Field, E. (2001). *The Virtual Score: Representation, Retrieval, Restoration*. MIT Press.

Holder, L. B. and Cook, D. J. (2009). Graph-based data mining. *Encyclopedia of Data Warehousing and Mining*, 2:943–949.

Holder, L. B., Cook, D. J., Djoko, S., et al. (1994). Substucture discovery in the SUBDUE system. In *KDD workshop*, pages 169–180.

Holtzman, S. R. (1980). A generative grammar definition language for music. *Interface*, 9(1):1–48.

Hsu, J., Liu, C., and Chen, A. (2001). Discovering nontrivial repeating patterns in music data. *IEEE Transactions on Multimedia*, 3(3):311–325.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Humphries, M. D. and Gurney, K. (2008). Network 'small-world-ness': A quantitative method for determining canonical network equivalence. *PLoS ONE*, 3(4):e0002051.

Jackendoff, R. and Lerdahl, F. (1981). Generative music theory and its relation to psychology. *Journal of Music Theory*, pages 45–90.

Jacobson, K., Fields, B., and Sandler, M. (2008). Using audio analysis and network structure to identify communities in online social networks of artists. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jehan, T. (2005). *Creating Music by Listening*. PhD thesis, Massachusetts

Institute of Technology, Boston.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *ECML '98 Proceedings of the 10th European Conference on Machine Learning*, pages 137–142.

Johnson, C. M. (2001). A survey of current research on online communities of practice. *The Internet and Higher Education*, 4(1):45–60.

Jordà, S. and Wüst, O. (2001). FMOL: A system for collaborative music composition over the web. *Proceedings of the 12th International Workshop on Database and Expert Systems Applications (DEXA 2001)*.

Jordan, B. and Henderson, A. (1995). Interaction analysis: Foundations and practice. *The Journal of the Learning Sciences*, 4(1):39–103.

Kaliakatsos-Papakostas, M. A., Floros, A., Kanellopoulos, N., and Vrahatis, M. N. (2012). Genetic evolution of L and FL-systems for the production of rhythmic sequences. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, pages 461–468, Philadelphia, Pennsylvania, USA.

Kapur, A. (2004). Query-by-beat-boxing: Music retrieval for the dj. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*.

Kartomi, M. J. (1990). *On Concepts and Classifications of Musical Instruments*. University of Chicago Press Chicago.

Karypis, G. (2002). CLUTO - A Clustering Toolkit. Technical Report #02-017, University of Minnesota, Department of Computer Science.

Karypis, G., Han, E.-H., and Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery. In *Proceedings IEEE International Conference on Data Mining (ICDM 2001)*, pages 313–320.

Lartillot, O., Eerola, T., Toiviainen, P., and Fornari, J. (2008). Multi-feature modeling of pulse clarity: Design, validation and optimization. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR 2008)*, pages 521–526.

Laurier, C. (2011). *Automatic Classification of Musical Mood by Content-Based Analysis*. PhD thesis, Universitat Pompeu Fabra.

Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral*

Participation. Cambridge University Press, 1 edition.

Lee, K. and Ellis, D. P. W. (2010). Audio-based semantic concept classification for consumer video. *IEEE Transactions on Audio, Speech & Language Processing*, 18(6):1406–1416.

Lessig, L. (2008). *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. Penguin Press HC, The.

Levin, T. Y. (2003). "Tones from out of nowhere": Rudolph Pfenninger and the archaeology of synthetic sound. *Grey Room*, pages 32–79.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development: Parts I and II. *Journal of Theoretical Biology*, 18:280—315.

Lourenço, B. F., Ralha, J. C., and Brandao, M. C. (2009). L-systems, scores, and evolutionary techniques. In *Proceedings of the 6th Sound and Music Computing Conference (SMC' 09)*, pages 113–118.

Madsen, S. T. and Jørgensen, M. E. (2003). *Automatic Discovery of Parallelism and Hierarchy in Music*. PhD thesis, Aarhus Universitet, Datalogisk Institut, Aarhus.

Mandler, J. M. (2000). Perceptual and conceptual processes in infancy. *Journal of Cognition and Development*, 1(1):3–36.

Martin, K. D. and Kim, Y. E. (1998). Musical instrument identification: A pattern-recognition approach. *The Journal of the Acoustical Society of America*, 104(3):1768–1768.

McCormack, J. (1996). Grammar based music composition. *Complex Systems*, 96:321–336.

McKinney, M. F. and Breebaart, J. (2003). Features for audio and music classification. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, volume 3, pages 151–158.

Merker, B. (2002). Music: the missing Humboldt system. *Musicae Scientiae*, 6(1):3–21.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. In Chen, C., editor, *Pattern Recognition and Artificial Intelligence*, volume 116, pages 374–388. Academic, New York.

Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, pages 522–536, Galway, Ireland.

Miletto, E. M., Pimenta, M. S., Bouchet, F., Sansonnet, J.-P., and Keller, D. (2011). Principles for music creation by novices in networked music

environments. *Journal of New Music Research*, 40(3):205–216.

Miletto, E. M., Pimenta, M. S., Hoppe, A. F., and Flores, L. V. (2009). Who are the web composers? In *Online Communities and Social Computing*, pages 381–390. Springer.

Miletto, E. M., Pimenta, M. S., Vicari, R. M., and Flores, L. V. (2005). CODES: a Web-based environment for cooperative music prototyping. *Organised Sound*, 10(3):243–253.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97.

Montuori, A. and Purser, R. (1995). Deconstructing the lone genius myth: Toward a contextual view of creativity. *Journal of Humanistic Psychology*, 35(3):69–112.

Moore, B. C. (2012). *An Introduction to the Psychology of Hearing*. Brill.

Moore, B. C. and Glasberg, B. (1996). A revision of Zwicker's loudness model. *Acustica – Acta Acustica*, 82(2):335–345.

Navas, E. (2012). *Remix theory: the aesthetics of sampling*. Springer, Vienna.

Newman, M. E. (2004). Analysis of weighted networks. *Physical Review E*, 70(5):056131.

Newman, M. E. J. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.

Noack, A. (2009). Modularity clustering is force-directed layout. *Physical Review E*, 79(2):026102.

Ong, B. and Streich, S. (2008). Music loop extraction from digital audio signals. In *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME)*, pages 681–684, Hannover.

Pampalk, E., Hlavac, P., and Herrera, P. (2004). Hierarchical organization and visualization of drum sample libraries. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, pages 378–383, Naples.

Pampalk, E., Rauber, A., and Merkl, D. (2002). Content-based organization and visualization of music archives. In *Proceedings of the 10th ACM International Conference on Multimedia (ACM-MM '02)*, pages 570–579.

Patterson, R. D., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C.,

and Allerhand, M. (1992). Complex sounds and auditory images. *Auditory physiology and perception*, 83:429–446.

Peeters, G. and Deruty, E. (2010). Sound indexing using morphological description. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):675–687.

Pfaltz, J. L. and Rosenfeld, A. (1969). Web grammars. In *Proceedings of the 1st international joint conference on Artificial intelligence*, pages 609–619. Morgan Kaufmann Publishers Inc.

Pons, P. and Latapy, M. (2005). Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer.

Prusinkiewicz, P. (1986). *Score generation with L-systems*. Ann Arbor, MI: MPublishing, University of Michigan Library.

Prusinkiewicz, P. and Lindenmayer, A. (1996). *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc., New York, NY, USA.

Rabiner, L. R. and Juang, B.-H. (1993). Fundamentals of Speech Recognition. *Book*.

Radcliffe-Brown, A. R. (1948). *The Andaman Islanders*. Cambridge University Press.

Ramirez, J., Górriz, J. M., and Segura, J. C. (2007). *Voice Activity Detection. Fundamentals and Speech Recognition System Robustness*. INTECH Open Access Publisher.

Ricard, J. (2004). *Towards Computational Morphological Description of Sound*. PhD thesis, DEA pre-thesis research work, Universitat Pompeu Fabra, Barcelona.

Roads, C. (1978). Composing grammars. In *International Computer Music Conference*.

Roads, C. and Wieneke, P. (1979). Grammars as Representations for Music. *Computer Music Journal*, 3(1).

Roberts, C. and Kuchera-Morin, J. (2012). Gibber: Live coding audio in the browser. In *Proceedings of the International Computer Music Conference (ICMC 2012)*, Ljubljana, Slovenia.

Rocchesso, D. and Fontana, F. (2003). *The Sounding Object*. Mondo estremo.

Rohrhuber, J. (2007). Network music. In Collins, N. and d'Escriván, J., editors, *The Cambridge Companion to Electronic Music*, pages 140–170. Cambridge University Press, Cambridge, 4th edition.

Roma, G. and Herrera, P. (2010a). Community structure in audio clip sharing. In *2010 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, pages 200–205.

Roma, G. and Herrera, P. (2010b). Graph grammar representation for collaborative sample-based music creation. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, pages 17:1–17:8, Piteå, Sweden.

Roma, G. and Herrera, P. (2013). Representing music as work in progress. In *Structuring Music through Markup Language: Designs and Architectures*, pages 119–134. IGI Global Snippet.

Roma, G., Herrera, P., Zanin, M., Toral, S. L., Font, F., and Serra, X. (2012a). Small world networks and creativity in audio clip sharing. *International Journal of Social Network Mining*, 1(1):112–127.

Roma, G., Janer, J., Kersten, S., Schirosa, M., Herrera, P., and Serra, X. (2010). Ecological acoustics perspective for content-based retrieval of environmental sounds. *EURASIP Journal on Audio, Speech, and Music Processing*, pages 1–11.

Roma, G., Nogueira, W., and Herrera, P. (2013). Recurrence quantification analysis features for environmental sound recognition. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–4.

Roma, G., Xambó, A., and Laney, R. (2012b). Factors in human recognition of timbre lexicons generated by data clustering. In *Proceedings of the Sound and Music Computing Conference (SMC '12)*.

Rosselet, U. and Renaud, A. B. (2013). Jam on: a new interface for web-based collective music performance. In *Proceedings of the 2013 Conference on New Interfaces for Musical Expression (NIME '13)*, pages 394–399.

Russolo, L. (1986). *The Art of Noises*. Pendragon Press, New York.

Salembier, P. and Sikora, T. (2002). *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York.

Schaeffer, P. (1966). *Traité des Objets Musicaux*. Seuil, Paris.

Schafer, R. M. (1977). *The Soundscape: Our Sonic Environment and the Tuning of the World*. Inner Traditions/Bear.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 32–36. IEEE.

Schwarz, D. (2006). Concatenative Sound Synthesis: The Early Years.

Journal of New Music Research, 35(1):3–22.

Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104.

Schwarz, D., Britton, S., Cahen, R., and Goepfer, T. (2007). Musical applications of real-time corpus-based concatenative synthesis. In *International Computer Music Conference (ICMC 2007)*.

Schwarz, D., Schnell, N., and Gulluni, S. (2009). Scalability in content-based navigation of sound databases. In *International Computer Music Conference (ICMC 2009)*, Montreal, Canada.

Serrà, J., Serra, X., and Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11.

Shannon, B. J. and Paliwal, K. K. (2003). A comparative study of filter bank spacing for speech recognition. In *Microelectronic Engineering Research Conference*, volume 41.

Shao, Y. and Wang, D. (2008). Robust speaker identification using auditory features and computational auditory scene analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 1589–1592.

Slaney, M. (1993). An efficient implementation of the Patterson-Holdsworth auditory filter bank. Technical report, Apple Computer, Perception Group, Tech. Rep.

Sloetjes, H. and Wittenburg, P. (2008). Annotation by category: ELAN and ISO DCR. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC '08)*, pages 816–820.

Smaill, A., Wiggins, G., and Harris, M. (1993). Hierarchical music representation for composition and analysis. *Computers and the Humanities*, 27(1):7–17.

Sohn, J., Kim, N. S., and Sung, W. (1999). A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3.

Tanaka, A., Tokui, N., and Momeni, A. (2005). Facilitating collective musical creativity. In *Proceedings of the 13th annual ACM International Conference on Multimedia (MULTIMEDIA '05)*, pages 191–198, Singapore.

Torsello, A. and Hancock, E. R. (2003). Computing approximate tree edit distance using relaxation labeling. *Pattern Recognition Letters*, 24(8):1089–1097.

Traunmüller, H. (1990). Analytical expressions for the tonotopic sensory

scale. *The Journal of the Acoustical Society of America*, 88(1):97–100.

Tsai, F. S. (2010). Review of techniques for intelligent novelty mining. *Information Technology Journal*, 9(6):1255–1261.

Tucker, R. (1992). Voice activity detection using a periodicity measure. *IEE Proceedings I (Communications, Speech and Vision)*, 139(4):377–380.

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

Uzzi, B. and Spiro, J. (2005). Collaboration and creativity: The small world problem. *American Journal of Sociology*, 111(2):447–504.

Van Den Doel, K., Kry, P. G., and Pai, D. K. (2001). FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 537–544.

van Grootel, M., Andringa, T. C., and Krijnders, J. (2009). DARES-G1: database of annotated real-world everyday sounds. In *Proceedings of the NAG/DAGA Meeting 2009*, pages 996–999.

Verspagen, B. and Duysters, G. (2004). The small worlds of strategic technology alliances. *Technovation*, 24(7):563–571.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Webber, C. L. and Zbilut, J. P. (1994). Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology*, 76(2):965–973.

Weinberg, G. (2005). Interconnected musical networks: Toward a theoretical framework. *Computer Music Journal*, 29(2):23–39.

White, H. D., Wellman, B., and Nazer, N. (2004). Does citation reflect social structure?: longitudinal evidence from the "Globenet" interdisciplinary research group. *Journal of the American Society for Information Science and Technology*, 55(2):111–126.

Whitehead Jr, E. J. and Wiggins, M. (1998). WebDAV: IEFT standard for collaborative authoring on the Web. *IEEE Internet Computing*, 2(5):34–40.

Wiggins, G., Miranda, E., Smaill, A., and Harris, M. (1993). A framework for the evaluation of music representation systems. *Computer Music Journal*, 17(3):31–42.

Wold, E., Blum, T., Keislar, D., and Wheaton, J. (1996). Content-based

classification, search, and retrieval of audio. *IEEE MultiMedia*, 3(3):27–36.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.

Worth, P. and Stepney, S. (2005). Growing music: Musical interpretations of L-Systems. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D. W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G. D., and Squillero, G., editors, *Applications of Evolutionary Computing*, volume 3449, pages 545–550. Springer.

Xu, M., Maddage, N., Xu, C., Kankanhalli, M., and Tian, Q. (2003). Creating audio keywords for event detection in soccer video. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo (ICME '03)*, volume 2, pages II–281.

Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2003)*, pages 721–724.

Zaki, M. J. (2005). Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae*, 66(1):33–52.

Zbilut, J. P. and Webber, C. L. J. (2006). Recurrence quantification analysis. In Akay, M., editor, *Wiley Encyclopedia of Biomedical Engineering*. John Wiley and Sons, Hoboken.

Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

Zhang, T. and Kuo, C.-C. (1999). Classification and retrieval of sound effects in audiovisual data management. In *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 730–734.

Zils, A. and Pachet, F. (2001). Musical mosaicing. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-01)*, volume 2, pages 730–734, Limerick, Ireland.

Appendix A: Publications by the author

Book Chapters

Roma, G., and Herrera P. (2013). Representing Music as Work in Progress. In: Steyn, J., ed. Structuring Music through Markup Language: Designs and Architectures. IGI Global, 2013, pp. 119–134.

Journal Articles

Bogdanov, D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., et al. (2014). ESSENTIA: an open source library for audio analysis. ACM SIGMM Records. 6(1).

Roma, G., Zanin M., Herrera P., S. L., Font F., and Serra X. (2012). Small world networks and creativity in audio clip sharing. International Journal of Social Network Mining (IJSNM), 2012, 1(1), pp. 112 - 127.

Roma, G., Janer J., Kersten S., Schirosa M., Herrera P., and Serra X. (2010). Ecological acoustics perspective for content-based retrieval of environmental sounds. EURASIP Journal on Audio, Speech, and Music Processing. (2010).

Janer, J., Finney N., Roma G., Kersten S., and Serra X. (2009). Supporting Soundscape Design in Virtual Environments with Content-based Audio Retrieval. Journal of Virtual Worlds Research, 2009, 2(3).

Conference Papers

Xambó, A., Roma G., Laney R., Dobbyn C., and Jordà S. SoundXY4: Supporting Tabletop Collaboration and Awareness with Ambisonics Spatialisation. In: Proceedings of the 14th International Conference on New Interfaces for Musical Expression, 2014. pp. 40 - 45.

Bogdanov, D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., et al. ESSENTIA: an Open-Source Library for Sound and Music Analysis. In: Proceedings of the ACM International Conference on Multimedia (MM'13), 2013. Font, F., Roma G., and Serra X. Freesound Technical Demo. In: Proceedings of the ACM International Conference on Multimedia (MM'13), 2013. pp. 411–412.

Roma, G., Nogueira W., and Herrera P. Recurrence Quantification Analysis Features for Environmental Sound Recognition. In: Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013.

Mayor O., et al. ESSENTIA: an Audio Analysis Library for Music Information Retrieval. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'13), 2013. pp. 493–498.

Font, F., Roma G., Herrera P., and Serra X. Characterization of the Freesound Online Community. In: Proceedings of the Third International Workshop on Cognitive Information Processing, 2012. pp. 279–284.

Roma, G., Xambó A., Herrera P., and Laney R. Factors in human recognition of timbre lexicons generated by data clustering. In: Proceedings of the Sound and Music Computing Conference (SMC2012), 2012.

Janer, J., Roma G., and Kersten S. Authoring augmented soundscapes with user-contributed content. In: Proceedings of the ISMAR Workshop on Authoring Solutions for Augmented Reality, 2011.

Akkermans, V., Font F., Funollet J., De Jong, B, Roma G., Togias S., et al. Freesound 2: An Improved Platform for Sharing Audio Clips. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

Janer, J., Kersten S., Schirosa M., and Roma G. An online platform for interactive soundscapes with user-contributed content. In: Proceedings of the AES 41st International Conference on Audio for Games, 2011.

Roma, G., and Herrera P. Community structure in audio clip sharing. In: Proceedings of the International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010), 2010.

Roma, G., and Herrera P. Graph grammar representation for collaborative sample-based music creation. In: Proceedings of the 5th Audio Mostly Conference, 2010. Schirosa, M., Janer J., Kersten S., and Roma G. A system for soundscape generation, composition and streaming. XVII CIM-Colloquium of Musical Informatics, 2010.

Roma, G., Janer J., Kersten S., Schirosa M., and Herrera P. Content-based retrieval from unstructured databases using an ecological acoustics taxonomy. In: Proceedings of the International Community for the Auditory Display (ICAD), 2010.

Janer, J., Haro M., Roma G., Fujishima T., and Kojima N. Sound Object Classification for Symbolic Audio Mosaicing: A Proof-of-Concept. In: Proceedings of the Sound and Music Computing Conference, 2009. pp. 297–302.

Roma, G., and Xambó A. A tabletop waveform editor for live performance. In: Proceedings of the 8th International Conference on New Interfaces for Musical Expression, NIME08, 2008.