UNIVERSITAT POMPEU FABRA

Musical expectation modelling from audio: a causal mid-level approach to predictive representation and learning of spectro-temporal events

Amaury Hazan

A dissertation submitted to the Department of Information and Communication Technologies at the Universitat Pompeu Fabra for the program in Computer Science and Digital Communications in partial fulfilment of the requirements for the degree of

—

Doctor per la Universitat Pompeu Fabra

Director de la tesi:

Doctor Xavier Serra
Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra, Barcelona

Tesi Doctoral UPF / 2010

This research was performed at the Music Technology Group of the Universitat Pompeu Fabra in Barcelona, Spain. This research was partially funded by the EmCAP project FP6-IST, contract 013123.

Acknowledgements

Nicolas Wack taught me that programming is not a burden, it's a lifestyle. Paul Brossier taught me that open source is not real-time, but when it's open-source and real-time it feels good. Perfecto Herrera taught me to take it easy, to make it solid and to make it happen. Xavier Serra gave me the opportunity to be part of MTG and to start this thesis. Noemí taught me not to work on weekends, even if models of listening are fun.

The researchers I met at MTG and thanks to MTG taught me many things too: many thanks to Bram de Jong, Gunter Geiger, Jens Grivolla, Thomas Aussenac, Julien Ricard, Alex Freginals, Xavier Forns, Xavier Amatriain, Alex Loscos, Jordi Janer, Inês Salselas, Fabien Gouyon, Hendrik Purwins, Maarten Grachten, Rafael Ramirez, Anssi Klapuri, Esteban Maestre, Alfonso Pérez, Emilia Gómez, Ricard Marxer, Piotr Holonowicz, Joshua Eichen, Owen Meyers, Eduard Aylon, Cyril Laurier, Jordi Funollet and Sergi Jordà.

My brother taught me about personal commitment, but he works on weekends. My parents told me that a thesis should be completed after six years, even if models of listening are fun.

Actually, they are.

Abstract

We develop in this thesis a computational model of music expectation, which may be one of the most important aspects in music listening. Many phenomenons related to music listening such as preference, surprise or emotions are linked to the anticipatory behaviour of listeners. In this thesis, we concentrate on a statistical account to music expectation, by modelling the processes of learning and predicting spectro-temporal regularities in a causal fashion.

The principle of statistical modelling of expectation can be applied to several music representations, from symbolic notation to audio signals. We first show that computational learning architectures can be used and evaluated to account behavioral data concerning auditory perception and learning. We then propose a *what/when* representation of musical events which enables to sequentially describe and learn the structure of acoustic units in musical audio signals.

The proposed representation is applied to describe and anticipate timbre features and musical rhythms. We suggest ways to exploit the properties of the expectation model in music analysis tasks such as structural segmentation. We finally explore the implications of our model for interactive music applications in the context of real-time transcription, concatenative synthesis, and visualization.

Resumen

Esta tesis presenta un modelo computacional de expectativa musical, que es un aspecto muy importante de como procesamos la música que oímos. Muchos fenómenos relacionados con el procesamiento de la música están vinculados a una capacidad para anticipar la continuación de una pieza de música. Nos enfocaremos en un acercamiento estadístico de la expectativa musical, modelando los procesos de aprendizaje y de predicción de las regularidades espectro-temporales de forma causal.

El principio de modelado estadístico de la expectativa se puede aplicar a varias representaciones de estructuras musicales, desde las notaciones simbólicas a la señales de audio. Primero demostramos que ciertos algoritmos de aprendizaje de secuencias se pueden usar y evaluar en el contexto de la percepción y el aprendizaje de secuencias auditivas. Luego, proponemos una representación, denominada *qué/cuándo*, para representar eventos musicales de una forma que permite describir y aprender la estructura secuencial de unidades acústicas en señales de audio musical.

Aplicamos esta representación para describir y anticipar características tímbricas y ritmos. Sugerimos que se pueden explotar las propiedades del modelo de expectativa para resolver tareas de análisis como la segmentación estructural de piezas musicales. Finalmente, exploramos las implicaciones de nuestro modelo a la hora de definir nuevas aplicaciones en el contexto de la transcripción en tiempo real, la síntesis concatenativa y la visualización.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Computational modelling of music aims at providing tools that enable us to analyze, understand and interact with music. In this context, computational modelling of music expectation is focusing on creating models of music listening which have the ability to represent musical events in a meaningful way. This representation gives the model a predictive capability about the future events to be heard. Both representation and prediction processes take place and evolve during listening in a causal, dynamic way.

Here, we start from the audio signal and view music as a sequence of acoustic events that are ordered through time to form musical patterns. As such, our definition does not focus on precise aspects of western music such as meter or harmony. However, this representation is flexible enough to accommodate a range of musical audio signals, from commercial music to casual sounds and voice onomatopoeia.

The role of computational models of music expectation is to create a meaningful representation of sequences of acoustic events and to learn the structure of those sequences through the generation of expected events. This makes our approach closely related to studies in music representation, perception, and sequential learning. Computational approaches to these phenomena have already been proposed in the past, however there are only a few approaches that tried to provide a bridge between real-world musical stimuli and models of expectation.

We suggest that computational models of music expectation have the potential to bring new approaches to interaction with musical content, because of their ability to represent a stream of music and to predict it. This opens the door to applications of real-time visualization of the musical structure, musical interaction with computers and musical gaming, to cite a few.

1.1 Motivation

Models of music listening have played an increasing role in the recent years. Music listening has proved to provide many insights about cognitive processes such as memory, attention or emotion. Music stimuli have been used to derive functional brain maps of music listening using neuroimaging techniques. This enabled to determine which cortical functions were important when listening to music, as compared to processing other stimuli such as speech or images. Researchers have proposed computational counterparts that attempt to simulate some of the phenomena involved in sound and music listening. Because of the multifaceted aspects of music listening, those models usually focus on a particular aspect of music perception (e.g. pitch processing) and can rarely process real-world music pieces. Nevertheless, those computational models have proved useful in allowing theories of music perception to be evaluated empirically through simulations.

On a more practical side, computer scientists have proposed new computer tools and interfaces to help music lovers to make, discover and share music. Some of these tools provide ways of extracting relevant information from musical audio tracks, which is called music content analysis and has been one of the major branches of the Music Information Retrieval (MIR) research (see (Downie, 2003) for an introduction). In the majority of cases, these models of music content analysis are loosely related to models of music perception. Works in the field of content-based Music Information Retrieval aim at analyzing, indexing and managing collections of music. Music content analysis engines produce compact summary of musical pieces in a bottom up approach. First, low level descriptors are computed along a musical signal. Then, these signals are averaged to form mid and high-level descriptors that represent the musical piece as a *signature*, that is, a compact summary of the piece.

Recent approaches are able to summarize various aspects of a musical piece such as rhythm (Gouyon and Dixon, 2005), tonality (Gómez, 2006), or spectral content (Wang, 2003). The resulting signature can then be used to perform queries among a the music collection, such as similarity, fingerprinting, recommendation or playlist generation. It is not unusual to qualify this approach as *bag-of-features* oriented: each item's signature is a bag of features that help describing it. In this process, the time structure of the musical information is often collapsed or reduced to a few measures describing the statistical properties of the signal features (e.g. average, standard deviation).

The general work flow of content-based MIR systems does not differ dramatically from other Information Retrieval systems that manage collections of images or text. In all these cases, approximations are made concerning how the listener -the reader or observer- perceives each document. Because features are averaged through time, one strong approximation that is made in such systems is the timing of perception. The perception of our environment is nevertheless a phenomenon that evolves through time. When observing a picture, subjects produce a sequence of eye saccades and thus track the points of interest of the picture. Readers discover a text one word

after the other and form a semantic representation which is modified and completed when subsequent words are read.

Similarly, listeners follow a musical piece *as it unfolds* through time. During this process, each new musical event is processed as a new evidence for appreciating the piece; At each point in time the listener could ask unconsciously: Did I perceived something? Does the sound I just I heard gives me a feeling of repetition? Does it surprise me? Is this sound, note or chord worth remembering? What feelings does it elicitate to me? The temporal dynamics of the listening process are driven by important cognitive functions such as attention, representation, memory, prediction, expectation and emotion (Peretz and Zatorre, 2005). In this thesis, we emphasize prediction and expectation as the dynamic processes that govern the timeline of music listening. At each point in time , based on what has been heard so far, the listener forms expectations about *what* is going to be heard and *when* it is going to be heard. Subsequent events are then compared to these expectations. In this context, the interplay of representation, expectation generation and comparison of incoming events with former expectations forms the temporal dynamics of the listening process. Jones and Boltz (1989) refer to this process as *future-oriented attending*.

Temporal dynamics are also crucial when it comes to musical performance and practice. In a band, music practitioners can constantly *follow* each other to produce a collective, synchronized, musical rendition. Non-trained musicians are able to follow a musical piece by tapping their feet or clapping their hands. If the musical piece is predictable enough, the prediction will be correct. If a sudden change occur the predictions will be wrong during a certain time lag until the structure can be followed again. This is a dynamic behaviour that we would like to be reflected using a computational model, and that can not be addressed with bag-of-features approaches.

Overall, our main motivation is to investigate under which conditions a computational model a music expectation can be built, define what type of expectations can be generated by the model, and explore new forms of musical interaction that would take advantage of such a computational model.

1.2 A theory of expectation

Researchers have attempted to define the cognitive functions elicited by music listening. Meyer (1956) highlighted the role of expectation in the music listening process. According to him, the listener's expectations provide an important tool in the process of composing a musical piece, and that musical emotions can arise from the interplay between composition and expectations. For instance, at each point in time, the listener's expectation can be fulfilled by the composer, making the structure easier to follow, or delayed, creating a feeling of tension. Huron (2006) refines these ideas and formulates a theory called ITPRA, an acronym for responses caused by Imagination, Tension, Prediction, Reaction, and Appraisal. For Huron, expectations in music and other domains arise from these five functionally distinct neurophysiological systems. Each system responds to stimulations from the other systems, and

the sequential ordering of these responses creates the overall listening experience. The five systems are defined by Huron as follows: "Feeling states are first activated by imagining different outcomes (*Imagination*). As an anticipated event approaches, physiological arousal increases, often leading to a feeling of increasing tension (*Tension*). Once the event has happened, some feelings are immediately evoked related to whether one's prediction were borne out (*Prediction*). In addition, a fast reaction response is activated based on a very cursory and conservative assessment of the situation (*Reaction*). Finally, feeling states are evoked that represent a less hasty appraisal of the outcome (*Appraisal*)." Huron shows that the ITPRA theory of expectation helps describing many aspects of music listening and musical organization in general, and backs Meyer by suggesting that musicians "have proved the most adept at manipulating the conditions of the different dynamic responses". These works provide a basis for understanding that the timing of perception influences how music is appreciated by listeners, and helps us defining the role of expectation.

The influence of expectation in the listening process can be seen as twofold: on the one hand, expectations can be formed in a bottom-up fashion, depending on low-level statistics of the auditory environment. This process may be regarded as largely automatic, the way it affects perception has been formalized into principles such as Gestalt (Narmour, 1990) or Auditory Scene Analysis (Bregman, 1990). In this context, Narmour suggested that certain aspects of music expectation are innate to music listeners rather than drawn from musical experience. On the other hand, expectations can originate from higher level processes, and be governed by a knowledge that has been built with the musical experience of each listener. Here, the listeners expectation can be influenced by learning and therefore depends on enculturation (Krumhansl, 1979; Hannon and Trehub, 2005). These higher-level expectations can in turn affect lower-level stage of music perception in a top-down fashion.

From a computational modelling perspective, these approaches to define expectation raise the following questions: What to expect? In other words, how do listeners represent musical events from an auditory stream such as music, in a way that they can form expectations on it?

1.3 Prediction-driven Computational Modelling

In his thesis, Ellis (1996) proposed a model of computational auditory scene analysis that is driven by the prediction of incoming events. As such, this model forms a milestone for building a general model of sound perception, because it addresses both issues of representation and prediction. By using a coarse representation of sounds - showing an emphasis on generality rather than precision, the system can simulate experiments dealing with real world street sounds. However, the lack of musically informed representations make the system impractical for representing adequately sounds in musical mixtures.

Representation of musical signals is a complex issue that has been later

addressed by content-based MIR systems. However, many works trying to build predictive models in music have relied on ad hoc, symbolic representations of music. In he last two decades, many approaches have been proposed to build symbolic models of music prediction, see (Bharucha and Todd, 1989; Todd and Loy, 1991; Mozer, 1994; Tillmann et al., 2000; Lartillot et al., 2001; Eck and Schmidhuber, 2002; Pearce and Wiggins, 2004). Indeed, putting aside the issue of representing the musical signal has enabled researchers to focus on other questions that arise when investigating prediction and modelling it: Is prediction an innate feature or is it learned? If so, how do we learn to predict? How good are we at prediction?

1.3.1 Sequential Learning

Most of our day-to-day activities involve sequencing of actions to achieve a desired goal, from sequencing words to form a sentence, to driving an automobile or following directions on a road map. Lashley and Jeffress (1951) has highlighted the ubiquity of sequentiality or serial order in our behavior. Sequential learning has been investigated under different perspectives, from neurophysiology to psychology to computational modelling. These works aimed at defining what part of sequential prediction was innate and what part was acquired. Several mechanisms that influence the prediction process have been identified, and they may be linked to conscious training. However, another mechanism called implicit learning suggested the hypothesis that subjects could learn and exploit the underlying structure of a sequence by mere exposure to a structured sequence of events.

Some issues researchers attempt to define are the kind of structure that can be learned, the capacity or sequential memory, or the amount of sequential context needed to perform accurate predictions. Speech, language, and music, due to their very sequential nature, have been considered as cases of sequential learning. Computational modelling studies have investigated the use of certain predictive models and compared them with behavioral data.

1.4 Expectation modelling in musical audio

We aim at developing a model of expectation that is able to form a representation of music from an audio stream, and generate expectations while "listening" to this stream. Subsequently we aim at defining how these expectation can be used by the system to support the listening process and to provide feedback to users of musical systems.

1.4.1 Goals

This PhD dissertation first discusses the theoretical and empirical foundations of music representation and expectation, and then proposes technical approaches to provide models and implementations that simulate these phenomena. The dissertation also stresses how the proposed approach can be integrated into musical system to provide novel functionalities and applications.

Our hypothesis is that expectation modelling provides an alternative approach of music listening modelling when it comes to work with real-world auditory streams, by shifting the paradigm of bag-of-features processing to a dynamic listening process. This thesis aims at showing why and how expectation modelling can be implemented in this context. The goals of the PhD dissertation are presented below:

- Review the theoretical, cognitive, and perceptual concepts involved in music listening. This will first lead us to review issues such as the representation of auditory and musical events. This review will then be complemented by behavioral data concerning musical sequences learning.
- Review computational models of sequential learning that have been used in the literature as well as models that have been specifically applied to music.
- Propose a framework for simulating and evaluating sequential learning experiments, and study the impact of music representation in these simulations.
- Propose a representation of musical audio signals based on the time dependencies (when) between acoustic units (what).
- Integrate a representation and an expectation module in order to build a model of music listening that can be applied to a range of musical audio signals.
- Validate the proposed expectation model, showing the implications of our approach and how it can be used in artificial music listening systems.

1.4.2 Application contexts

The application contexts of this work are summarized below:

- Novel description of musical content in MIR systems: expectation modelling that takes into account the timing of music listening provides a complementary description of musical excerpts. The predictability of the musical content provides cues describing the structure of musical signals. As such, expectation-driven modelling can provide complementary accounts of musical complexity, and may be used for performing segmentation of musical content.
- Synthesis of prediction: By coupling the external audio-based representation and the internal representation of the musical structure, it is possible to create an auditory rendition of the system's predictions. This opens the door to interesting music applications that can complement existing mosaicing techniques (Schwarz, 2004; Jehan, 2005) but also provides a means to inspect the system's internal representation of the auditory stimuli.
- Interaction with music: An online model of music listening and expectation can analyze music *as it is produced*. This enables applications involving musical interaction with users: the users can perform music and get auditory and visual feedback from the musical system. Interaction is not limited to music practitioners. The system works with

audio signals, and nonmusicians can also interact musically, like in casual games.

1.5 Summary of the PhD work

The goal of this PhD is to contribute to clarify how a causal music expectation system can be applied to real-world, audio signals.

Models of music expectation have traditionally been applied to ad-hoc symbolic representation of musical events. By starting from this traditional approach to expectation modelling, we use a symbolic representation of musical events in which case the task of an expectation model is to predict *what is coming next*.

Using this setting, we define a computational framework that enables to simulate experiments of learning of tone sequences that use the forced-choice task paradigm (Saffran et al., 1999). Our simulations show that the ability to reproduce behavioral data is largely influenced by the representation that is chosen to describe musical events.

These findings lead us to define a model in which the representation of musical events and their expectation is linked. We propose a representation of the acoustic musical stream that takes into account both acoustic properties of musical sounds (*what*) and the time in which they occur (*when*). This leads us to define a range of schemes that combine these characteristics, which form the basis of the *what/when* expectation model. We show that different representation models of the acoustic events -either supervised or unsupervised- can be integrated with the *what/when* expectation model.

The system is then evaluated using a range of musical signals containing percussive sounds, monophonic melodies, or mixtures from commercial recordings. The results suggest that fully unsupervised representation models can represent and track musical signals that can be described in a monophonic way, whether supervised representation models are needed to tackle the polyphonic representation of more complex sound mixtures.

Then, we show how this expectation model can be integrated in audio-processing musical systems, to enable real-time interaction, synthesis of prediction and visualization of both musical structure and expectation dynamics.

1.6 Structure of this document

The organization of this PhD dissertation is the following. In Chapter 2 we give an account of auditory and music perception from multiples perspectives with the goal to illustrate how prediction has been studied and modelled. Music perception is viewed as an active process that involves both representation and prediction of musical events, and we aim at defining how these aspects can be combined in a simplified listening model. The musical information that can be extracted when attending a musical stream, such as timbre, melody or rhythm, is presented. In our review, we use contributions from (Purwins et al., 2008). Then, we report several experiments that inves-

tigate how learning of musical sequences takes place. Those works provide methods for assessing how well a set of musical stimuli can be learned by subjects and investigate which musical structure can be learned and what musical factors influence learning.

In Chapter 3 we introduce computational methods of sequential learning and present past approaches to music prediction using symbolic representations.

In Chapter 4, we develop a learning simulation framework in which well-known prediction methods are used to simulate a behavioral experiment focusing on statistical learning of tone sequences, and stress that the choice of the musical representation influences learning and prediction. This chapter reports the findings published by Hazan et al. (2008). The issue of computational representation of musical events is developed in Chapter 5, with an emphasis on description of musical audio signals and prediction-driven approaches to audio analysis.

In Chapter 6, we integrate the representation and prediction layers. We introduce our representation and expectation models, which are aimed at generating expectations while analyzing audio signals by maintaining an internal symbolic representation of the musical stream in terms of time and acoustic properties of musical events. We provide a set of metrics for characterizing the predictive behavior of the system, evaluate our model using different sets of musical excerpts, and discuss the results obtained. This chapter uses and extends the models and findings presented by Hazan et al. (2009).

In Chapter 7 we show how the *what/when* expectation model can be integrated into musical systems for providing analysis of musical signals or for real-time applications that involve interaction and visualization.

Finally, in Chapter 8 we present the general conclusions of this dissertation and suggest future work directions.

CHAPTER 2

Context of Research

2.1 Chapter Summary

We introduce in this chapter a number of musical dimensions that may be considered when focusing on expectation of musical sequences. We will present these dimensions according to two perspectives: accounts of music perception that are rooted in western music, and more general accounts of auditory perception and psychoacoustics. When considering music expectation, we first need to define which musical events or auditory objects can be considered, as well as the temporal structures that organize those objects in musical sequences. On top of addressing the issue of music representation, we aim at modelling expectation through learning of musical sequences. Therefore, in a second part, we will propose a review of auditory and music sequence learning experiments, show the methodology employed to characterize how learning takes place, and show which factors influence the process of learning musical sequences.

2.2 Describing the dimensions of music: Timbre, Melody and Rhythm

Which musical dimensions should we take into account when considering music expectation? Because music perception is a complex phenomenon, which can be described at various scales, this question is difficult to answer. In our approach, we aim at describing music as a sequence of auditory events which are organized through time. In this simplified view, auditory events can consist of notes, sounds, or attacks. We will first review approaches to describe these auditory events sequences, and will then review how to describe the temporal structures that organize these events.

2.2.1 What to expect - expectation of musical elements

When listeners listen to a musical passage, they direct their attention and process the successive acoustic cues in a way that makes them able to identify musical events such as notes, chords, voices, percussive strokes, etc.

In other words, listeners extract auditory objects from the stream they pay attention to, some with a better precision than others. For instance, when listening to a string section, it can be difficult or even impossible for a expert to identify how many instruments are playing together, and at which time is located each instrument attack. Conversely, some auditory objects such as a hand clap may be easier to identify and locate in time by non-musicians.

Auditory objects

The question of how we process sound to organize it into meaningful perceptual *objects* has been at the center of Bregman's investigations. Representing sound objects in the auditory environment is mandatory in one's everyday life, for instance to locate incoming cars while crossing the street in a noisy environment. However, those auditory objects do not necessarily refer to concrete sound sources, rather the refer to the mental representations we create from our acoustic environment. The task of creating and maintaining those auditory objects is called Auditory Scene Analysis (ASA, (Bregman, 1990)). Three ASA key processes are *segmentation*, *integration*, and *segregation*. Probably the best-known example of segregation is the cocktail party problem, in which individuals are able to segregate one particular voice among many other voices and sounds. Integration of sounds takes place when different sounds are associated together to form a sound unit. This happens, for instance, when individual notes are grouped together and identified as a chord, or when a succession of chords is perceived as a musical color. When listening to the everyday sound environment or a musical piece, the listener often hears a mixture of acoustic components and has to identify from this mixture a representation that makes sense by isolating independent *streams*. Several factors influence how streams are segregated from mixtures of sounds, an important aspects being the temporal ordering of events in the mixture.

Top-down and bottom-up processing

Top-down and bottom-up processing form two pathways involved in music perception, and have been considered in computational models. First, bottom-up processing starts from the input waveform, and successively combines the low level information to create more abstract cues that can be used as input for higher-level auditory objects. Bottom-up processing is also referred to as *data-driven* processing. Bottom-up processing is common practice in Music Information Retrieval systems, where low-level descriptors are extracted from the signal (see (Peeters, 2004) for a review of such descriptors). These low-level descriptors are then combined to obtain higher-level information, such as sound category, genre, or key. For instance, sounds

with high transients and short sustain will be associated more likely to percussion or speech plosives than to bowed strings. Such account of music perception through the bottom-up representation of auditory objects has been developed (Schaeffer, 1966; Bregman, 1990; Roads, 2004).

Conversely, top-down processing starts from an internal representation of the environment and prior knowledge regarding the auditory objects, their regularities and co-occurrences. The information flows down, by successively comparing a higher level representation with lower level sensory input, and adapting the higher level model to reflect the sensory input. Top-down processing governs several aspects of auditory perception such as auditory restoration, where listeners can still identify recordings of spoken words where syllables have been deleted or replaced with noise bursts (Warren and Warren, 1970), plays an important role in resolving the cocktail party problem (Bregman, 1990), and enable listeners to identify a specific auditory stream for a complex sound mixture. Top-down processing also helps understanding the *context effect*: in a listening experiment using speech, Ladefoged (1989) has shown that listeners identify the same auditory stimulus at the end of a sentence as two different words depending of the beginning of the sentence.

Overall, it should be noted that bottom-up and top-down processes should be viewed as complementary cognitive processes (Bregman, 1990). The model we will introduce in Chapter 6 provides an approach to integrating both top-down and bottom-up processes, by limiting our definition of musical sequences to a set of acoustic units that are governed by temporal patterns. Acoustic properties of sound attacks and timing information are integrated to form a set of timbre and temporal symbols in bottom-up fashion. Conversely, musical structures can be learned and expectations can be generated in a symbolic fashion and mapped backed into the auditory domain in a top-down fashion. Following, we introduce the main musical dimensions, that is, timbre, melody and rhythm, we will refer to in this dissertation.

Timbre Perception

One of the characteristics that help distinguish between different strokes, notes or voices is timbre. The definition of timbre is still subject of controversy for the lack of agreement in the literature to the point that it has been qualified as "the psychoacoustician's multidimensional wastebasket category for everything that cannot be qualified as pitch or loudness" (McAdams and Bregman, 1979). Timbre is first a subjective sensation that depends on a variety of acoustic properties? According to Grey (1977),"a major aim of research in timbre perception is the development of a theory for the salient dimensions or features of classes of sounds." There is some consensus that the envelope of sounds as well as their spectral content affect the timbre of a sound.

Among the physical quantities than may influence timbre, Schouten (1968) lists the following:

1. The range between tonal and noiselike character.
2. The spectral envelope.
3. The time envelope in terms of rise, duration, and decay.
4. The changes both of spectral envelope (formant-glide) and fundamental frequency (micro-intonation).
5. The prefix, an onset of a sound quite dissimilar to the ensuing lasting vibration

Further works have attempted to characterize the perceptual dimensions of timbre by performing listening experiments. Grey (1977) proposed a characterization of timbre in three dimensions. In this study, subject are presented with sounds from different instruments but whose fundamental frequency remains constant. The listeners rated the similarity between this sounds and the results were analyzed using Multi-Dimensional Scaling (MDS) and related them to properties of the signal. This study has then been followed by (Wessel, 1979; Krumhansl, 1989; McAdams et al., 1995), in which the sound used and the experimental setup have been refined. Krumhansl (1989) uses three perceptual dimensions, namely *attack quality*, *spectral flux*, and *brightness*. In (McAdams et al., 1995), the authors proposed a perceptual characterization of timbre in three dimensions along with a computational definition of physical quantities (i.e. spectral centroid, rise time, spectral flux) that match these perceptual dimensions. The three-dimensional space obtained is this study is shown in Figure 2.1.

Melody Perception

From a cognitive perspective, melody perception concerns primarily perceptual grouping. This grouping depends on relations of proximity, similarity, and continuity between perceived events. As a consequence, what we are able to perceive as a melody is determined by the nature and limitations of perception and memory. Melody perception presumes various types of grouping. One type of grouping divides simultaneous or intertwined sequences of pitches into *streams*. The division into streams is such that streams are internally coherent in terms of pitch range and the rate of events. A second type of grouping concerns the temporal structure of pitch sequences. A subsequence of pitches may form a *melodic grouping* (Snyder, 2000), if preceding and succeeding pitches are remote in terms of pitch register, or time, or if the subsequence of pitches is repeated elsewhere. Such groupings in time may occur at several time scales. At a relatively short time scale the groupings correspond to instances of the music theoretic concepts of *motifs*, or *figures*. At a slightly longer time scale, they may correspond to *phrases*. According to Snyder (2000), the phrase, which is typically four or eight bars long, is the largest unit of melodic grouping we are capable of directly perceiving, due to the limits of short term memory. Rather than being fully arbitrary, (parts of) melodies are often instantiations of *melodic schemata*, frequently

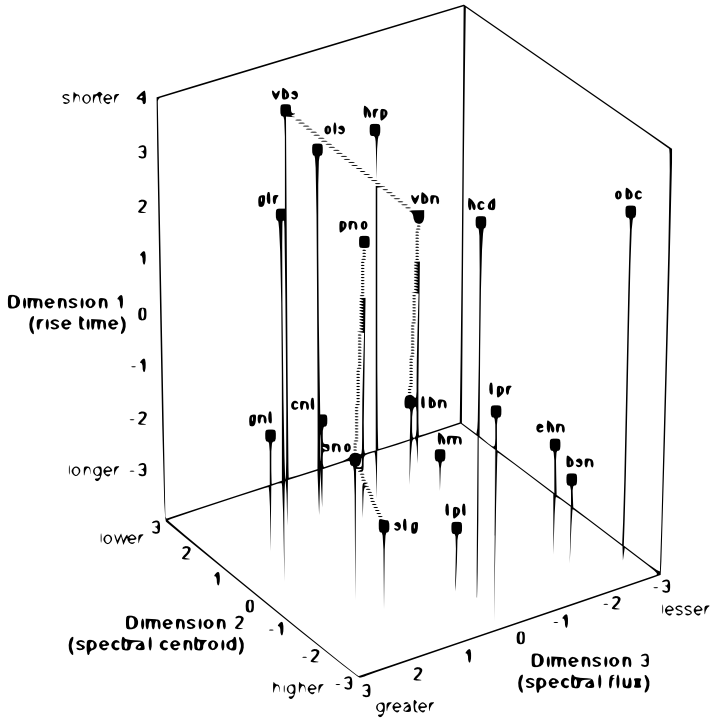


Figure 2.1: Three dimensional timbre space characterized by McAdams et al. (1995). Each point represent an instrument timbre label.

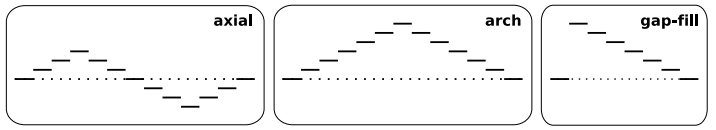


Figure 2.2: Pitch contour diagrams of three melodic schemata: axial, arch, and gap-fill, from (Snyder, 2000).

recurring patterns of pitch contours. The most common melodic schemata are *axial* forms, *arch* forms, and *gap-fill* forms (Meyer, 1956). Axial forms fluctuate around a central pitch, the 'axis'; Arch forms move away from and back to a particular pitch; And gap-fill forms start with a large pitch interval (the 'gap') and continue with a series of smaller intervals in the other registral direction, to fill the gap (Snyder, 2000). The pitch contours of these schemata are illustrated in Figure 2.2.

The Implication-Realization model

The most well-known model for melodic expectancy is the Implication-Realization (I-R) model (Narmour, 1990, 1992) The (I-R) model is a theory of perception and cognition of melodies. The theory states that a melodic

musical line continuously causes listeners to generate expectations of how the melody should continue. The nature of these expectations in an individual are motivated by two types of sources: innate and learned. According to Narmour, on the one hand we are all born with innate information which suggests to us how a particular melody should continue. On the other hand, learned factors are due to exposure to music throughout our lives and familiarity with musical styles and particular melodies. According to Narmour, any two consecutively perceived notes constitute a melodic interval, and if this interval is not conceived as complete, it is an *implicative interval*, i.e. an interval that implies a subsequent interval with certain characteristics. That is to say, some notes are more likely than others to follow the implicative interval. Two main principles recognized by Narmour concern *registral direction* and *intervallic difference*. The principle of registral direction (PRD) states that small intervals imply an interval in the same registral direction (a small upward interval implies another upward interval and analogously for downward intervals), and large intervals imply a change in registral direction (a large upward interval implies a downward interval and analogously for downward intervals). The principle of intervallic difference (PID) states that a small (five semitones or less) interval implies a similarly-sized interval (plus or minus 2 semitones), and a large interval (seven semitones or more) implies a smaller interval.

Based on these two principles, melodic patterns or groups can be identified that either satisfy or violate the implication as predicted by the principles. Such patterns are called structures and are labeled to denote characteristics in terms of registral direction and intervallic difference.

For example, the P structure ("Process") is a small interval followed by another small interval (of similar size), thus satisfying both the PRD and the PID. Similarly the IP ("Intervallic Process") structure satisfies the PID, but violates the PRD. Some structures are said to be retrospective counterparts of other structures. They are identified as their counterpart, but only after the complete structure is exposed. In general the retrospective variant of a structure has the same registral form and intervallic proportions, but the intervals are smaller or larger. For example, an initial large interval does not give rise to a P structure (rather to an R, IR, or VR, see figure 1, top), but if another large interval in the same registral direction follows, the pattern is a pair of similarly sized intervals in the same registral direction, and thus it is identified as a retrospective P structure, denoted as (P).

Figure 2.3 (left) shows eight prototypical Narmour structures. A note in a melody often belongs to more than one structure. Thus, a description of a melody as a sequence of Narmour structures consists of a list of overlapping structures. The melody can be parsed in order to automatically generate an implication/realization analysis. Figure 2.3 (left) shows the analysis for a melody fragment. As pointed out by Grachten et al. (2006), The I-R analysis can be regarded as a moderately abstract representation of the score, that conveys information about the rough pitch interval contour and, through the boundary locations of the I-R structures, it includes metrical and durational information of the melody as well. It is worth noting here that the I-R model rely on perception principles (proximity, similarity, closure) that are

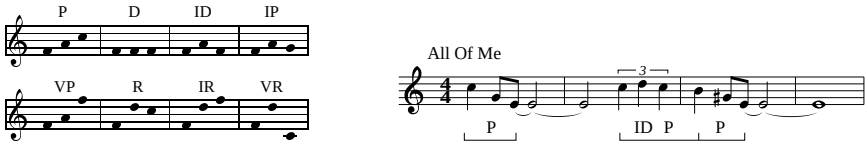


Figure 2.3: Eight of the basic structures of the Implication-Realization (I-R) model (*left*). First measures of *All of Me* (Marks & Simons 1931), annotated with I/R structures (*right*). From (Grachten et al., 2006).

not specific to melody. In this context the I-R model might be adapted to process sequences of auditory objects other that pitches (e.g. non-pitched percussive events).

2.2.2 When to expect: expectation of time structures

In parallel to forming expectations about *what* is going to be heard, the listener also anticipates *when* the events will be heard. The interaction of what and when expectation form the main basis of predictive listening in music. A representation of events in time has to be used to anticipate the timing of future events. Different representations of time that may be useful to form such expectations. We will present in this section some key concepts to understand these representations.

Firstly, periodic auditory cues such as clock ticks are easy to predict, because the period of the cue can be processed by listeners. This is not the case for events in which the period between cues is either too short (successive events will be perceived as a whole or as independent streams) or too long, in which case the perception of periodicity vanishes. Demany et al. (1977) sustains the existence of a preferred tempo spectrum ranging from 60 to 120 beats per minute anchored approximately at 100 beats per minute (we refer to London (2002), who provides a detailed review of cognitive constraints on beat perception). When a musical excerpt is attended, listeners are able to follow the most salient periodic pulse and tap in time with the music. The notion of *tactus* is associated to this most salient pulse.

From Periodicity to Rhythm

Meter represents a finer-grained description of the temporal structure which enables to describe musical events in a hierarchical way. As noted by Tillmann (2008), "temporal regularities include the organization of event-onset-intervals through time leading to a sensation of meter - a sensation of a regular succession of strong and weak beats superimposed over an isochronous pulse. Temporal regularities also include the temporal patterns of onset intervals creating rhythms that are perceived against the metrical background". As Huron (2006) points out, meter may also be seen from a prediction-driven perspective. Huron points out that “Meter provides a recurring temporal template for coding and predicting event onsets.”. If a sequence of inter-onset intervals has a regular period, “the temporal expec-

tations might be represented using mental oscillators of various frequencies and phases". Then, by considering where the onsets are located along the sequence period, the listener may be able to locate the onset moments that are more likely than others. This gives rise to a hierarchy of temporal events "which can be expressed in terms of their metric position within a recurring temporal template". To this extent, meters can be viewed as predictive schemas that enable expectation of temporal events. The time expectation model we propose in Chapter 6 aims at learning the time regularities between acoustic objects. We will suggest that the prediction statistics of such regularities can give rise to a rough, implicit representation of meter.

On a more precise scale, a musical sequence can be described in terms of rhythm. Rhythm is a musical concept with a particularly wide range of meanings, and as such it is essential to delimit the scope of what we will be talking about when discussing rhythm. An all-encompassing description of rhythm would be that it is about the perception of the temporal organization of sound. As far as we consider *musical* sound, rhythm in this sense is tightly connected with the concept of meter. Several studies have even questioned the separation between the processes of meter and rhythm perception altogether (Hasty, 1997; Rothstein, 1981; Drake, 1998; Lerdahl and Jackendoff, 1983).

Although rhythm is ambiguous and could, as in the bottom-line definition given above, also include meter, a common use of the words rhythm and meter is expressed in a paraphrase of London (2006): "Meter is how you count time, and rhythm is what you count–or what you play while you are counting". When we hear a sequence of events that can be located in time, periodicities in the timing of these events will cause in us the awareness of a *beat*, or *pulse*. By this, we mean a fixed and repeating time interval with a particular offset in time, that can be thought of as a temporal grid, forming a context in which the perceived events take place. The perception of events often makes some beats feel stronger, or more accented, than others, giving rise to a hierarchical grid of beats.

As a context for the perception of musical events, meter serves to categorize events in terms of temporal position and accent, and inter-onset intervals (IOI) of events in terms of duration. A common interpretation of *rhythm* is the grouping of sequences of such categorized events, where groups often contain one accented event and one or more unaccented events (Cooper and Meyer, 1960). This accounts for the influence of meter on the perception of rhythm. The perception of actual musical events on the other hand also affects the perception of meter. Firstly, in actual performances phenomenal cues like the loudness and duration of events are used to accent particular metrical positions, thereby suggesting a specific meter (Lerdahl and Jackendoff, 1983; Palmer, 1989). In addition to this, the frequency distribution of musical events over metrical locations is mainly correlated with meter, rather than, for example, musical style or period (Palmer and Krumhansl, 1990). This indicates that frequency distribution of events in a piece could be a perceptual clue to determining meter. Palmer and Krumhansl (1990) also show that listeners appear to have abstract knowledge of the accent structure of different kinds of meters, which become more fine-grained as a

result of musical training. They suggest that the accent structures of different meters are learned by virtue of their ubiquity in Western tonal music.

A first step towards the understanding of rhythm perception is the study of perceived event durations. Differentiation of event durations happens in an early stage of auditory processing. Several factors are known to affect the perceived durations, leading to a difference between the physical duration and the perceived duration of an event. These include the phenomenon of auditory streaming, intensity/pitch difference between evenly spaced notes, and metric context. Likewise, perceived inter-onset intervals are influenced by event durations, and the different durations of successive events in their turn can affect the intensity and/or loudness perception of these (Terhardt, 1998).

Listening to two successive acoustic events gives rise to other cognitive processes than listening to each event separately. The judgment of successive events can be divided into two steps. The first step follows a modified version of Weber's law, in which the just-noticeable difference between two successive durations is proportional to their absolute length plus a constant of minimal discrimination (Allan, 1979). The second step consists in comparing the two durations. If they are similar, the second duration is related to the first one. If the two durations are considered very different they will both be related to previously established perceptual categories of durations (Clarke, 1989). Following, we will present several experiments that investigate how the quantization of rhythm takes place in listeners.

Rhythm quantization

Desain and Honing (2003) present two experiments to analyze the formation of perceived discrete rhythmic categories and their boundaries. The notion of rhythmic categories is interesting because those discrete categories refer to rhythms defined by continuous durations. In the first experiment they study the phenomenon of rhythm categorization, which consists in the mapping from the performance space –the set of possible performed IOI durations– to a symbolic space of rhythm representation –the set of duration categories, or nominal durations. An example comparing these two representations is shown in Figure 2.4.

Such a mapping expresses how much the IOI's of a performed rhythmic pattern can deviate from the nominal IOI durations that define the pattern, before it is being perceived as a different rhythmic pattern. The deviation from the nominal IOI durations is called *expressive timing*. Listeners tend to simplify durational relations assigned to an expressively timed pattern of onsets. Here, high consistency in the listener's judgment is predominant if the durational ratios are simple. Figure 2.5 shows the *time clumping* map that can be extracted from this experiment, by associating perceived rhythmic categories to triplets of performed inter-onset intervals.

In the second experiment, Desain and Honing study how the metrical context affects the formation of rhythmic categories. As a context, an acoustic pattern is provided by tapping a triple, duple, and simply a bar structure. Then the stimulus is played. Given triple and duple meter contexts, the

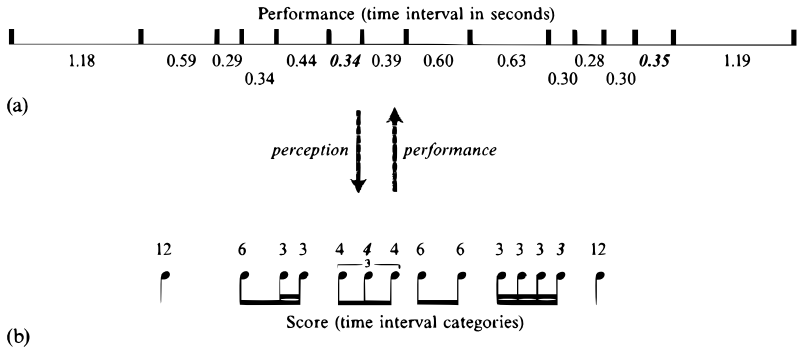


Figure 2.4: Examples of the two representations of time in music. A performed rhythm in continuous time (a) and a perceived rhythm in discrete, symbolic time (b). By Desain and Honing (2003).

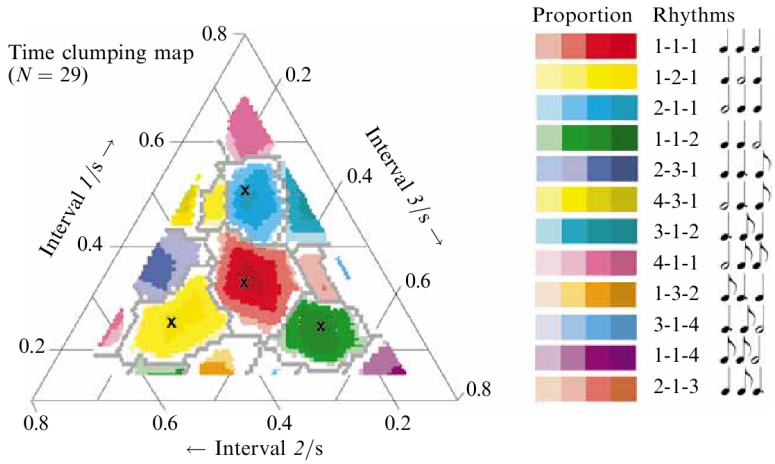


Figure 2.5: *Time clumping* map obtained after a listening experiment. The three triangle axes indicate the performance time as inter-onset intervals between successive strokes. The colored region refer to the symbolic categories that were chosen by the participants. Darker colors indicate a greater agreement between participants. Grey lines mark boundaries between rhythm categories. White regions indicate regions of low agreement among participants regarding the perceived rhythmic category. From (Desain and Honing, 2003).

stimulus is identified as having the meter of the context stimulus. If only bars are sonified, the stimulus is as well identified as duple meter in the majority of the cases.

As noted by Hannon and Trehub (2005), when listeners have categorized durations, they will continue to interpret new temporal patterns in terms of the formed categories, even in the presence of perceptible temporal changes

(Clarke, 1987; Large, 2000).

We retain from this overview that both performance and perceptual representations of time may be considered when representing timing in music. This gives us a basis to consider the representation of time from two viewpoints: an absolute, precise and sensor driven representation (referred to as "performance" time by Desain and Honing (2003)) and a compact, discrete representation of time symbols. To this extent, we suggest a model that may be able to manage both representation should implement mechanisms to translate the time information between these representations.

Rhythm along several acoustic dimensions Going back to figure 2.4 we observe that the rhythm stimulus that is considered is made of a single class of stroke. For the sake of simplicity, the only absolute information that is considered is the duration between successive strokes. We could argue that many popular rhythms (think about a rock drum performance) are using a more extended set of strokes (e.g. tom, cymbal, etc ...). The individual characteristic of each stroke (accent, instrument) provides additional information that may be used to encode the musical rhythm. For instance, being able to segregate the tom strokes form the cymbal strokes enables to track the stroke dependencies between events of the same acoustic category. This is an example of how top-down processing can influence lower-level representation layers. As we will see in Section 2.3.3, the acoustic similarities between strokes and notes can affect the way a musical sequence is learned. This will also form a basis for the computational model we propose in Chapter 6.

2.3 Learning and music perception

So far we have reviewed aspects of music perception that enable to define a musical stream under various modalities (pitch, timbre, time and rhythm). In the context of this dissertation, we are interested in investigating the role of expectation and its interplay with the representation with musical events. As we stressed in the introduction, this interplay can take place at several levels of abstraction. First, low-level expectation processes take place in a unconscious, automatic, hardwired fashion, and are reflected in Auditory Scene Analysis and Gestalt approaches. Then, higher-level expectation processes may be the result of a knowledge of the musical environment, that is acquired through experience. Several works have shown that an implicit knowledge about tonal structures can be acquired by mere exposure to tonal music and will be presented below. More generally, this section introduces works that aim to answer how we learn musical sequences and what musical aspects influence learning.

Apart from the hypothesis raised and results obtained, we will introduce the methodology used to understand the experimental setup employed as basis for the simulation framework we will introduce in Chapter 4. One major difficulty here is that the music we listen to on a daily basis, the "real-world" music, is usually too complex to be used in behavioral experiments where few parameters are to be varied, while many other need to be fixed, to obtain significant conclusions. However, we will show that these studies converge towards a common learning principle which takes place along several dimensions of music.

2.3.1 Implicit learning of auditory and musical regularities

In the 1960s, Reber pioneered the use of artificial grammars to investigate how these grammars could be learned by human subjects. Reber's work focused on the mechanism of implicit learning which suggests that the common structure of stimuli can be learned by means of mere exposure to those stimuli. To demonstrate this, the author generated sequences of letters using a finite state automaton (FSA) as shown is Figure 2.6. Sequences generated using this FSA share a common underlying structure, even if they differ in size. Participants were then presented novel sequences and asked if these sequences were grammatical or not. Reber measured how well these sequences could be memorized by subjects and showed that the results significantly outperformed those obtained with randomly generated sequences (Reber, 1967).

This phenomenon of implicit learning takes place independently of the modality of the visual stimuli (colors, letters, shapes). Subsequent works have translated these findings into the auditory domain. In this case, the sequences of letters are replaced by sequences of sounds. Tones are used by Altmann et al. (1995); Saffran et al. (1999); Loui et al. (2006).

In (Bigand et al., 1998a), the sequences are made of sounds with distinct timbres, whereas they consist of speech phonemes in (Saffran et al., 1996). Those studies have also employed various methods to generate exposure se-

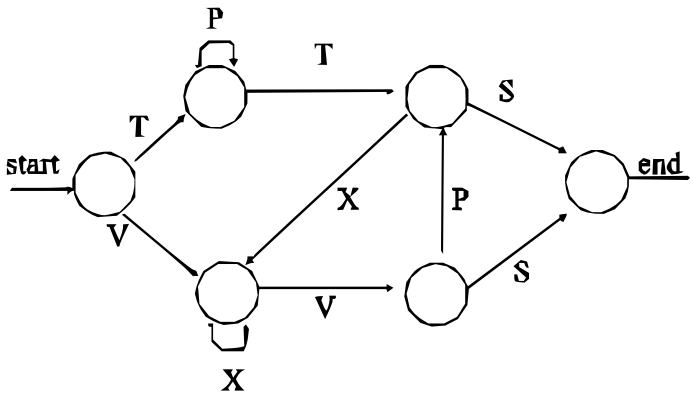


Figure 2.6: Finite-state automaton used by Reber (1967) to generate letter sequences.

quences. The underlying structure to be learned from exposure sequences can be derived from transition probabilities (Saffran et al., 1996, 1999; Tillmann and McAdams, 2004), finite-state automata (Loui et al., 2006), or grammars (Bigand et al., 1998b). Among those works, there is a common agreement that the structure of auditory sequences can be learned by mere exposure.

Therefore, investigating music perception from an implicit learning perspective may provide a complementary account to the music perception theories presented in Section 2.2, that do not focus explicitly on the learning process. We will provide in Chapter 4 a modelling study that shows how prediction and implicit learning are related. To achieve this, we need to understand the methodology employed to highlight implicit learning in musical sequences.

Learning statistical triplets in speech and tone sequences

Saffran et al. (1996, 1999) focused on assessing whether humans can learn regularities related to the transition probabilities regulating the elements inside auditory units (called words) or the word transitions in auditory material. In (Saffran et al., 1996) the auditory material was made of synthesized speech syllables, while in (Saffran et al., 1999) the authors used tone sequences, preserving the previous experimental setup. The authors created a set of artificial stimuli by setting high inside-word and low across-boundaries transition probabilities. In this work, two languages L1 and L2 were created. Each one contained 6 tone triplets, called tone-words. First, a random sequence of words of the defined language was presented to the subjects. However the tone triplets were presented in a regular order. There was no explicit cue indicating the boundaries among them. This means that the presented material appeared as a stream of tones which could be only

segmented using the statistical regularities of words. In the first experiment, words from L1 were *non-words* in L2, and vice versa. That is, there was no word in one language that appeared, even partly, in the other language. In the second experiment, words from one language were *part-words* in the other language, that is, only one tone differed between each language word. After exposure, the subjects had to perform forced-choice tasks involving exhaustive word-pairs belonging to each language. The task consisted in choosing which word of the pair had been effectively heard in the presented material. This study pointed out that the subjects were able to categorize above chance the words belonging to the material they were exposed. This means that the subjects are able to segment the input stream into words and to distinguish if a word presented subsequently belongs to the sequence they have been exposed to, and suggests that an automatic learning mechanism that exploits the co-occurrences of tones takes place.

Non western music scales

Loui et al. (2006) investigated the emergence of statistical regularities based on the presentation of non-western tonal sequences, generated using using the the Bohlen-Pierce scale (Mathews et al., 1984). Indeed, western listeners are mostly exposed to music using the western scale, so using a non-western scale to assess tone-word learning enables to discard the effect of enculturation in the learning process. In (Loui et al., 2006), the tone sequences were derived from two distinct finite state grammars, with some constraints (they respect the Narmour principle of closure). This work investigates whether the participants can acquire aspects of the structure of these two grammars using (a) forced-choice recognition and generalization (b) pre and post-exposure probe tone ratings, and (c) preference ratings. The authors introduced two experimental settings: in the first one a few exposure melodies were presented, while in the second one the number of exposure melodies (following a unique exposure grammar) was multiplied by three. As a general conclusion, the participants in experiment 1 tend to recognize better the melodies they have been exposed to while some in experiment 2 (the ones exposed to grammar 1) are also able to recognize new melodies generated from the grammar they have been exposed to (i.e. generalization). Furthermore, the post-exposure probe tone ratings were more correlated with the statistical distribution of the stimulus notes. Overall, the works presented above show that subjects are able to identify whether novel sequences respect or violate the structure of the exposure sequences.

This suggests that implicit learning also applies in auditory sequences under several modalities. One question arises from this: can the several dimensions that define a musical sequence have an influence on each other in way that makes it easier or harder to learn this sequence?

Learning statistical regularities in timbre sequences and the influence of timbral similarity

Tillmann and McAdams (2004) investigated the relation between acoustical similarity and statistical regularities in auditory sequences generated using different instruments. The authors proceeded by creating three sets of stimuli. The authors defined *statistical timbre triplets* by analogy with Saffran's work. In the first set, the statistical regularities were supported by acoustical similarities, i.e. the "timbral distance" between inner-word items was low while it was high across word boundaries. In the second set, the acoustical similarity contradicted the statistical regularities of the stimuli. Finally, in a third set, the timbral distance between consecutive events was neutral with respect to the statistical regularities. As a result, the authors found than subjects could learn timbre triplets from the first set with a higher accuracy than using the other sets. The worst learning accuracy was achieved using material where the acoustical cues were contradicting the statistical regularities of the auditory sequence. This work showed that the timbre information contained in auditory sequences influences the recognition of statistical units. Acoustic cues that support the statistical structure of timbre sequences facilitate the learning process. When the timbre information contradicts the same statistical structure, learning becomes more difficult.

2.3.2 Learning non-local dependencies

So far, we have considered how learning takes places for auditory sequences where adjacent events share a common structure. From this viewpoint, tones, sounds or phonemes are organized so that the transitions between successive events define this structure. However, musical sequences can also exhibit non-adjacent dependencies. To understand these non-local dependencies, let's consider a musical example. A melody is defined as a succession of notes that may have a complex structure, if we try to define this structure in terms of transitions between adjacent events. However, this apparently complex sequence can reveal two simpler interleaved melodies, such as a bass line and a main melody line. Therefore, we are interested in knowing how implicit learning takes place in a context of non-local dependencies. Past research on sequential order has shown that implicit learning of non-local patterns was generally difficult and that learners could learn patterns of temporally adjacent elements much better that they could learn non-temporally adjacent patterns (Cleeremans and McClelland, 1991; Cleeremans, 1993).

If we go back to the example of two interleaved melodies, the intuition would suggest that the bass line and main melody lines can be segregated by the listener because of the acoustical features that are common to each line. This acoustical features could include loudness, pitch or the characteristic timbre of the instrument used. If segregation can be successfully achieved (e.g. the bass line and main melody are perfectly identified), the problem is reduced to learning the local dependencies between elements of each stream.

Conversely, if errors are made during segregation (e.g. two streams are segregated, but both streams consist of interleaved notes from the bass and melody lines), the local dependencies between elements of each stream would not exhibit a clear structure. Further research has attempted to define to what extent acoustical cues can affect the learning of non-local dependencies.

2.3.3 Influence of acoustical cues

Newport and Aslin (2004) investigate how learning of non-local dependencies take place for speech stimuli, and report a similar study focused on tone sequences in (Creel et al., 2004). In both cases they propose a set of experiments focused on learning temporally non adjacent patterns and assessed whether acoustical cues could influence learning. The stimuli used in these experiments are similar to those used in (Saffran et al., 1999), but in this case the transition probabilities are defined between non-adjacent items. Two sets of items sharing are created, each one governed by its own transition probabilities, and the items of each set are then interleaved to form the exposure sequences. In the first experiment, the subjects are presented an exposure sequence where the items from both sets share the same acoustical features. Hence, apart from the nonlocal dependencies that govern each set, there is nothing that can help discriminating items from one set or the other. Acoustic cues are then introduced and applied on items from each set. In the case of tone sequences, items from a given set are rendered one octave lower (experiment 2), with a a sharply different timbre (experiment 3), or with moderately different timbre (experiment 4). In the case of speech, sequences of vowels (respectively consonants) are created, each item of the resulting sequence is then interleaved with unrelated consonants (respectively vowels).

The authors find similar results in both studies. They first confirm with the experiment the inability of learning highly consistent nonlocal dependencies in absence of acoustical cues, whether learning of local dependencies can be achieved successfully, as is was shown in (Cleeremans, 1993). Then, they show with the subsequent experiment that if acoustical cues are used to induce dissimilarities between the two sets, subjects can learn the structure of each set and therefore learn the dependencies between nonadjacent items.

The works presented above highlight the interplay between statistical dependencies - using low or high order context- and acoustical cues taking place when learning auditory sequences. Non-local dependencies cannot be learned without the help of acoustical cues that help the listener to segregates streams. This suggests that a model of acoustic sequence learning may take advantage of the events acoustic information and form streams as a preliminary process to sequence learning. We will get back to this point when exploring alternative expectation models applied to audio signals in Chapter 6.

2.3.4 Learning time dependencies

In the studies mentioned above, what matters is the ordering of auditory events and the acoustical cues that are used. However, they do not take into account the time structures (we refer to Section 2.2.2) that govern the occurrence of acoustic events. Compared to the amount of studies investigating implicit learning in auditory sequences, only a few works have validated, with behavioral experiments, the issue of how learning of time structures takes place. Boltz (1993) has shown that temporal dependencies can support the process of melody recognition. Hannon and Trehub (2005) investigate how rhythm structures are learned and the influence of cultural bias to learning. Indeed, rhythm patterns make possible to link remote events into a single stream, enabling their co-occurrences to be learned to some extent. The authors use two sets of rhythms to be learned, the first set has a simple metrical structure (temporal dependencies can be expressed with simple ratios such as $1/2$), while the second group use complex rhythms. Using a forced-choice task setup, they asked participants to classify novel sequences as preserving or violating the exposure sequences. The results shown that North-American adults could learn to recognize the simple rhythms but were unable to learn the complex rhythms while Bulgarian or Macedonian adults-whose musical culture makes a more frequent use of complex rhythms- could perform both tasks. This enculturation bias was confirmed with a last experiment, where North American infants were shown to be able distinguish structure-preserving complex rhythms from structure-violating sequences. This suggests that the musical environment North American infants are exposed to during development has overridden their initial ability to apprehend complex rhythms, because this ability is not useful in their environment.

2.3.5 Towards a statistical account of music perception

The works we have presented here focus on the various dimension we use to define musical sequences, i.e. timbre, pitch, and time. Independently of the modality used, these studies show that the common principle of implicit learning holds, that is, that mere exposure to sequences of events enables to learn to represent the structure of these sequences. Furthermore, the interplay between temporal and acoustical cues can enable to learn non-local dependencies, which would be too difficult to learn otherwise. More work has to be done to be able to deal with the complexity of real-world music. As Tillmann (2008) states, "Experimental research on music perception has to find a balance between the complexity of real musical material and strong experimental control of the used material leading to the use of simple tone contexts or melodies". However, despite the actual limitations of these works, we aim at integrating the general results obtained here in a computational model of music expectation.

Therefore, we will focus on building a model that can acquire a representation of musical structure through exposure, according to the principle of statistical learning. Our first motivation is to create a “blank” listener model guided by representation and learning principles. In this approach, the envi-

ronment will induce an internal model that will serve as a basis for generating expectations, instead of attempting to hard-wire a specific and static music theory in the model. The key questions that arise at the point are: what are the computational architectures that can be used to implement statistical learning systems and how can such architecture be evaluated. This questions will be answered in the two forthcoming chapters, where we will first define a range of models that can perform causal learning of sequential structure, and propose a use-case for evaluating such models based on Saffran et al. (1999). Furthermore, our review has shown that statistical learning may take place along several musical dimensions (e.g. timbre, melody, rhythm) and showed that learning may be facilitated when considering several dimensions instead of focusing on one. Because we aim at processing audio excerpts, we will propose, in Chapter 6, a set of representations of the audio stream in terms of acoustic cues and timing events, and investigate how these two dimensions can be combined.

CHAPTER 3

Causal models of sequential learning and applications to music

3.1 Chapter Summary

In this chapter, we introduce models of sequential learning and present computational approaches that have been used to simulate this phenomenon. As such, this chapter introduces the modelling formalism that will be used in this dissertation. We give an emphasis to *causal* models, that is, model whose predictions only depend on past observations, because causality provides a natural framework for modelling sequential learning. Furthermore, to reflect the existing work on sequence learning modelling and music sequence learning in particular, most of the approaches we present here are either symbolic approaches or are employed to process sequences represented in a symbolic way.

3.2 Models of sequential learning

A important core of computational techniques have been developed in the last 50 years to perform sequence analysis and prediction tasks. The spectrum of methods ranges from periodicity analysis (such as autocorrelation, time-frequency methods such as Fourier or wavelet transform), to artificial neural networks and probabilistic approaches. Predictive models have been used as forecasting application in many fields such as industry, finance or medicine (Box et al., 1976). Here, we focus on modelling learning as a dynamical process that evolves through time. This is why we make a distinction between causal and non-causal techniques, as explained below.

3.2.1 Causal versus batch processing

We aim to distinguish techniques that rely on a entire sequence to parse it from techniques that build and maintain a representation of the sequence accessed so far without being able to look ahead in the future. The former techniques are referred to as *batch* techniques - they operate on a batch of data from the beginning to the end, while the latter approaches are referred to as *causal* approaches. Formally, a causal system, is a system whose state at a specific instant t_0 depends only on the input x_t for values of t less or equal than t_0.

3.2.2 Markov-chain models

In mathematics, a Markov-chain is a stochastic process which respects the Markov property Markov (1971). This means that future states depend only on present states and are independent of past states. That is, the definition of the present state fully defines the evolution of future states. Formally, A Markov chain is a sequence of variables X_1, X_2, X_3, ...,X_n,X_{n+1} so that

$$Pr(X_{n+1}|X_n, ..., X_3, X_2, X_1) = Pr(X_{n+1}|X_n) \tag{3.1}$$

A variation of this is the Markov chain of *order* N, where N is finite. The state of this process can be written down as:

$$Pr(X_{n+1}|X_n, ..., X_3, X_2, X_1) = Pr(X_{n+1}|X_n, X_{n-1}, ..., X_{n-N}) \tag{3.2}$$

Here, N acts as a memory of fixed size. Hence, a Markov chain of *order* N is a process in which the future state X_{n+1} is fully defined by a fixed number of past observations. This assumption is the basis of many learning models that base their predictions on a past context of fixed size.

3.2.3 N-gram modelling

N-gram modelling provides a straightforward and efficient approach to probabilistic reasoning and expectation. This approach fits well to symbolic sequence modelling approaches, where each sequence item belongs to a fixed set of possible symbols. The basic idea is to construct tables of transition probabilities from subsequences of $N-1$ items to the next item by *counting* their occurrence in a set of training data.

We illustrate this by taking an example from MacKay (2003), focusing on modelling transition probabilities between letters (including the space character) in English sentences. Here, we consider the simple case where $N = 2$, also referred to as bi-gram modelling. Let us consider the random variable X, which corresponds to the outcome of a letter in the English corpus. The outcomes of X can be any letter of the alphabet, plus the space character. Similarly, we can define a random variable Y, which corresponds to the outcome of the following letter. Based on the training corpus, the first step consists in counting the occurrence of each single letter. By normalizing these counts, one can derive the marginal probability distribution of letters,

denoted $P(X)$, as shown in Figure 3.1a. In a similar way, one can count all the sequences of two letters that occur in the training corpus, and derive the joint probability distribution, denoted $P(X, Y)$. Figure 3.1b shows in a graphical way the probability distribution of such bi-grams. Here, the probabilities are represented as white squares in a two-dimensional table and sum to one. Based on the probabilities of occurrence of single letters and bi-grams, we can then derive the transition probabilities between two consecutive letters by computing the conditional probability between a letter given the outcome of the previous one:

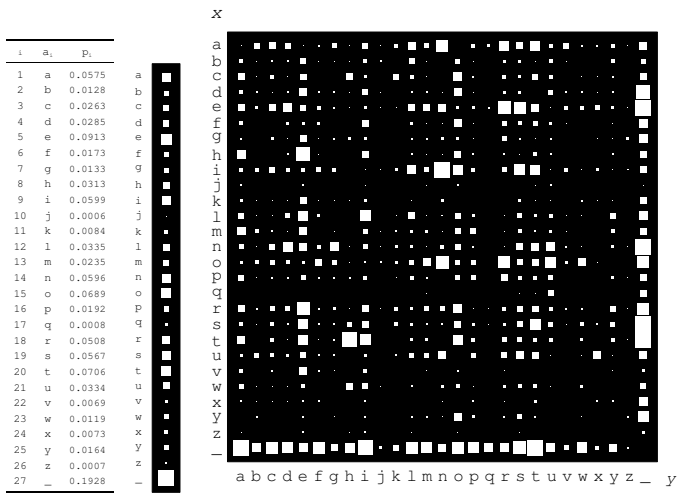
$$P(Y|X) = \frac{P(X,Y)}{P(X)} \tag{3.3}$$

Similarly, it is possible to derive the conditional probability of a letter given the outcome of the following one.

If we return to our example, the resulting transition probabilities for the English text corpus are shown in Figure 3.2. As an outcome of this process, the transition probability tables can now be used for prediction and expectation generation task. Indeed, when parsing a novel English sentence, it is possible, at each point in time, to obtain the condition probability distribution of the next letter.

We can now define N-grams as a function of the order parameter N. When a new event x_t is appended in the input sequence, a two-dimensional table, indexing the occurrences of all possible sequences of length n, is incremented by one at the position with row corresponding to the sequence x_{t-N+1} ... x_{t-1} and at the column corresponding to the symbol x_t. This means the count table has $nsymbols^N$ cells, where $nsymbols$ is the number of possible symbols. As such, this technique is expensive in terms of space, even if we can use sparse matrix implementations to reduce the size of the transition tables. From the count table and an observation of the actual sequence, we can derive the posterior probability for each possible symbol and produce an expectation accordingly. The N parameter controls the amount of past events considered in the prediction and has to be adjusted carefully. Indeed, an inaccurate choice of N may affect the behavior of the learner by biasing it to too general predictions (low N), and to over fit prediction (high N). Extensions have been proposed to combine predictions made from different context sizes.

Overall, the example we presented here gives a overview of the different processes involved in N-gram modelling. However, this example is also misleading , because it starts from the analysis of corpus of English text, derives knowledge from it, and then enables to perform prediction. As a matter of fact, there no need of a training corpus nor to separate learning and prediction. N-gram models can be applied in a causal way to symbolic sequences, by constantly updating their probability tables and by providing prediction based on the current tables. This way of coupling learning and prediction processes may precisely serve as basis for tackling the mechanism of implicit learning in the context of musical processing. Here, the continuous exposure to structured musical streams will substitute the corpus-based training shown in the English letters example.



(a) (b)

Figure 3.1: (a) Probability distribution of single letter $P(X)$ and (b) bi-grams probabilities $P(X, Y)$ derived from counting their occurrences in a corpus of English text. Adapted from MacKay (2003).

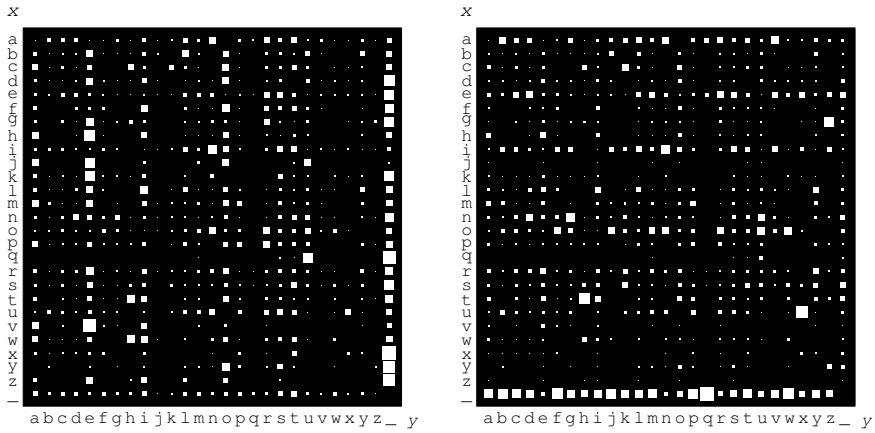


Figure 3.2: (a) Conditional probabilities of letter given the previous one $P(Y|X)$ and (b) conditional probability of a letter given the next one $P(X|Y)$ (b). These transition probabilities are derived from the single letter and bi-gram probabilities shown in Figure 3.1. Adapted from MacKay (2003).

Variable order markov-chain approaches

Extensions to markov chain models of fixed order have been proposed in the context of compression algorithms research. First, Incremental Parsing (IP, Ghezzi and Mandrioli (1979)), an online method based on compression theory. Another approach is based on Prediction Suffix Trees (PST, Allauzen et al. (1999)). Whereas a a markov chain of order M can be represented as a tree where all branches are of size M, in PST the size of the tree branches can be lower or equal to the maximum model order, which reduces the memory requirements of the model. Finally, an extension of N-grams, called Predictive Partial Match (PPM, Cleary and Witten (1984); Moffat (1990)), enables to combine transition tables of different context sizes. PPM will be presented in more details in Chapter 6.

3.2.4 Artificial Neural Networks

Artificial neural networks (ANN) form a branch of mathematical models that has been motivated by the structure and functional organization of biological systems. They consist of groups of interconnected artificial neurons that process information in a distributed way. Each artificial neuron is a simple unit that has its own activation state and is connected to other units via synaptic weights. The state of a given unit is determined depending of activations and weights of incoming units and the activation function of this unit. Synaptic connection can be either directed or undirected. For the sake of simplicity, we start introducing the typical case of feedforward neural networks, and will derive from this architectures that are able to model sequences of events. There are, however, more complex alternatives to this approach, which we will present in section 3.2.4. First, we show in Figure 3.4 how an artificial neuron is represented.

The activation function of each unit is arbitrary and does not need to be linear, which make ANN capable of non-linear transformation of data. Sensory inputs can be fed to the network and mapped to a useful representation which depends of the task considered. Formally, the incoming weighted inputs to a artificial unit j are first summed:

$$net_j = \sum_{i \neq j} w_{ij} x_i \tag{3.4}$$

The output of the unit j is then computed by summing a bias term θ_j and applying the unit activation function ϕ:

$$o_j = \phi(net_j + \theta_j) \tag{3.5}$$

The activation function ϕ can be either identity (we refer to linear units) or a thresholding function that ensures the unit output will be constrained to a certain range. Nonlinear thresholding functions such as the sigmoid or hyperbolic tangent are commonly used (Bishop, 1995).

Network Topology

The artificial neuron units are interconnected to form a topology. We consider here a typical data modelling system, in which sensory data is connected to a first layer of units, transformed along several hidden layers and mapped into a range of output units. Feedforward neural networks are made of several layers. Units from the same layer are not connected but receive inputs from the previous layer and project into the next layer, as shown in the example in Figure 3.5.

In this case, when presenting an input to the network, it is straightforward to compute a forward pass, which propagates the units activations from the input layer to the outputs.

ANN as an online adaptive learning system

A key characteristic of ANN is that network weights can be adapted using update rules during a learning phase. The weight update may depend on the inputs, outputs, external labels, or internal network state. This makes ANN an adaptive data modelling tool that can be used in range of supervised or unsupervised learning tasks such as classification, regression, or dimensionality reduction. The most popular learning rule for performing supervised training is called backpropagation (Werbos, 1974; Rumelhart and McLelland, 1986). In a supervised setting, an input vector is fed to the input, and the units activations are updated using the forward pass. Then the network's outputs are compared to the expected outputs (which forms the supervised data). An error function is used to perform this comparison, leading to a modification of the network weights in order to reduce this error, as explained below. Depending of the task considered, different types of output units and error functions are used, because they they enable to easily compute the error derivatives. We refer to Bishop (1995) for a justification of these measures.

Regression problems An error measure is chosen to compare output and targets. This measures depends on the task considered and has to be associated to a specific choice of output units. For regression tasks, linear output units are used and the most widely used error function is the sum-of-squares error, given by:

$$SSE = \frac{1}{2}\sum_{i=1}^{M}(o_i - t_i)^2 \tag{3.6}$$

where o_i is the activation of output unit i, t_i is the target value of output unit i, and M is the dimensionality of both vectors.

Classification problems As mentioned by Bishop (1995), it is possible to enforce a neural network to treat inputs or outputs as probabilities, which enables to train the networks and evaluate them according to probabilistic criteria. In a classification task, we would like to feed the network a vector

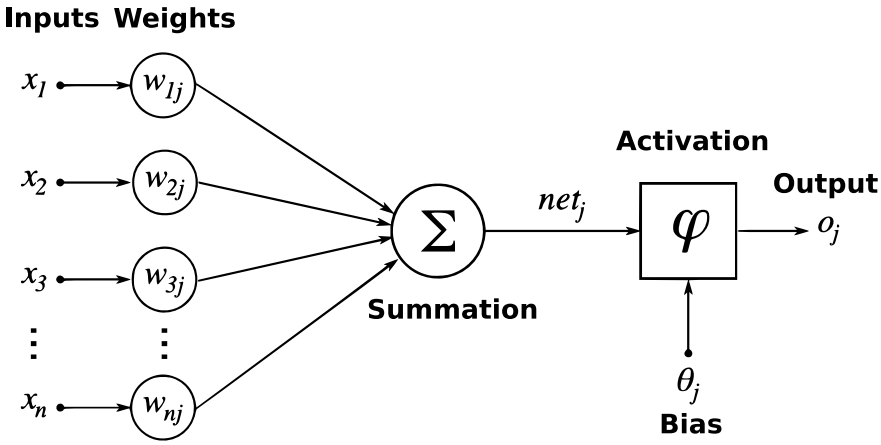


Figure 3.3: A typical artificial neuron unit. Incoming units are denoted $x_1, ..., x_n$. They are connected through synaptic weights denoted $x_{1j}, ..., x_{nj}$, where j is the index of the actual unit. The incoming weighted activations are summed along with a bias term θ_j. A possibly non-linear activation function is then applied to this sum to produce the unit output.

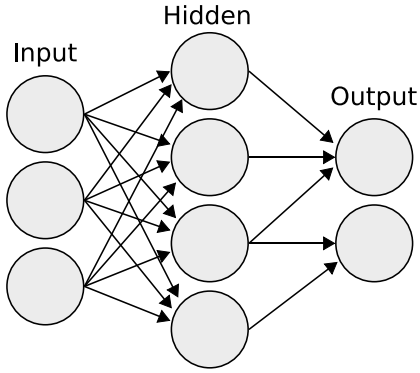


Figure 3.4: A typical feedforward neural network, also called multi-layer perceptron. The number of hidden layers can vary.

of sensory inputs and obtain the probability distribution over the possible classes. Here, a suitable error function for comparing the network's outputs with targets would be the cross-entropy error. If we consider two discrete probability distributions p and q over state space of dimension M, the cross-entropy between p and q is defined as:

$$H(p,q) = -\sum_{M} p(x)log(q(x)) \tag{3.7}$$

For classification tasks, a common choice is to use a *softmax* output layer, which ensures the output units activations sum to one. We consider an output unit of index j among M output units. Thus the output layer represent a multinomial distribution over N possible values. To compute the output of this unit, we replace Equation 3.5 by:

$$o_j = \frac{e^{net_j}}{\sum_{i=1}^{M} e^{net_i}} \tag{3.8}$$

Backpropagating the error Once the error between outputs and targets is computed, the networks weights need to be adjusted to minimize the error. Backpropagation is a computationally efficient algorithm that enables to evaluate the error-function derivatives of each unit with respect to its incoming weights, by going backward from the output units to the input units. When error derivatives corresponding to a unit are known, its incoming weights can be modified , thus enabling learning. We refer to Bishop (1995) for a full derivation of the backpropagation algorithm.

Batch versus online training Training can take place in two settings, batch and online: in the batch setting, the system processes a number of input samples as a batch. The mean error between outputs and expected targets is computed, allowing to adjust the weights using backpropagation. In the online setting, each time an input is presented, the networks outputs are compared to the expected outputs and backpropagation is applied. While the batch approach requires less computational efforts, it has two main disadvantages. The first drawback is that the network can be more likely trapped in a local minima during learning, because the average error signal provide less information than the individual errors. The second drawback is that batch processing does not allow to represent the timeline of sequential learning, when a system randomly initialized starts generating predictions, which are refined through exposure. Therefore, we focus here on the online variant of the backpropagation learning rule.

Modelling sequences of events and generating expectation

So far, we have considered networks in which, at each point in time, an input is applied and propagated to the outputs. In order to represent sequences the networks needs to encode the inputs temporal context. In our case, this would enable to build sequence prediction models. At each point

in time, the network would process the most recent input. Based on the input context, the network would be trained to predict the next item. Even if backpropagation is generally considered as a supervised technique, here we do not need to provide external labels: each time a new item is presented to the network, it is compared with the last network's prediction, which enables to backpropagate the error signal. Two simple modifications can be done to the feedforward neural network to handle sequence processing.

Time Delay Neural Network (TDNN) can be applied to *next-event prediction* tasks by using as many inputs as necessary to encode a fixed number of past events and using the output to predict the next event (Rumelhart and McLelland, 1986; O'Reilly and Munakata, 2000; Kuhn and Dienes, 2008). The number of past events encoded in the input layer determines the context available to provide a prediction, that is the context size is fixed explicitly to N past events, in a way similar to N-gram modelling. At each time step, the most recent input is appended and the past inputs are shifted using a delay line, the oldest input being not considered anymore. We show a three-layers Temporal Delay Neural Network with context of size N in Figure 3.5.

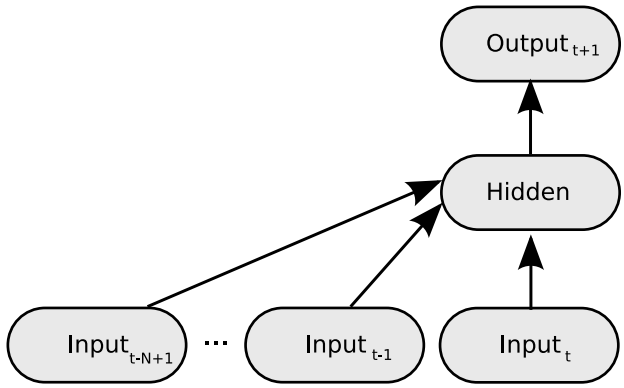


Figure 3.5: A temporal delay neural network with two layers, processing inputs over a context of size N. The individual artificial units are not shown. We show pools of artificial neurons with rectangles. Arrows show all-to-all feedforward connections between pools.

Simple Recurrent Network (SRN) are another variation that allows to represent past inputs in an implicit way, called Simple Recurrent Networks (Elman, 1990; Cleeremans and McClelland, 1991). A three-layer network is used, with the addition of a set of "context units" in the input layer, as shown in Figure 3.6. There are connections from the hidden layer to these context units fixed with a weight of one. The fixed back connections result in the context units always maintaining a copy of the previous values of the hidden units (since they propagate over the connections before the learning rule is applied). Thus the network can maintain a state, allowing it

to perform such tasks as sequence-prediction that are beyond the power of a standard feed-forward neural network.

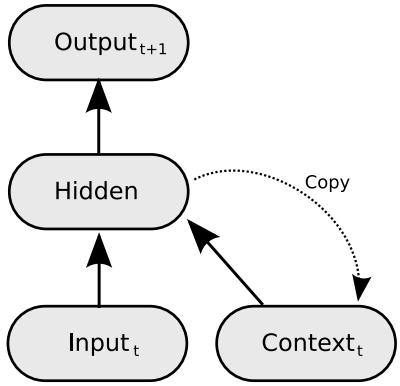


Figure 3.6: A simple recurrent network, obtained by appending a context layer to the network. At each time step, the hidden layer activation is computed depending of both input and context layer using feedforward connections. Then, we perform a copy of the hidden layer activations into the context layer. We show pools of artificial neurons with rectangle. Arrows show all-to-all feedforward connections between pools.

Simple Recurrent Networks have already been used in several works in order to build computational models of sequence learning from a cognitive perspective. Elman (1990) shows the ability of SRN to learn the structure of sequences made of numbers, letters or words. By analyzing the the network's test prediction error, Elman shows a trained SRN can perform word segmentation from sequences of letters. In Figure 3.7, we show the test prediction error of a SRN trained on a set of English sentences. The SRN prediction error is the highest at the beginning of a word (because the transition between two words is unpredictable) and decreases with subsequent word letters. Botvinick and Plaut (2006) use a variation of the SRN to investigate how learning of ordered lists of items takes place. Applications that are specific to music modelling are presented in Section 3.2.5

Other neural network approaches for sequence modelling

There have been many alternative models proposed for modelling sequence learning. Here we have shown that the SRN architecture implements recurrence using a copy of the hidden layer into the context layer, however the SRN can still be trained as a feedforward network, which makes the training straightforward with standard techniques. Fully recurrent neural architecture have been proposed, in which there can be backwards connection between layers as well as lateral connection between layer units or self-connections between artificial neurons. Specific learning rules have been devised for training these networks, the most notable being Real Time Recurrent Learning (Williams and Zipser, 1989) and Backpropagation Through

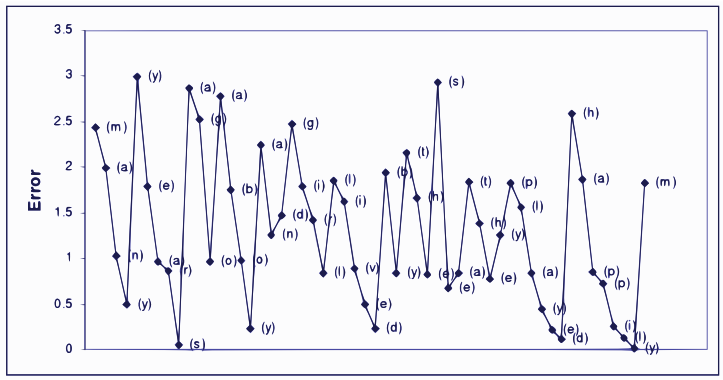


Figure 3.7: A word learning simulation presented by Elman (1990) using a SRN predictor. The trained prediction error is plotted along time. The letters presented at each point in time are shown in parentheses.

Time (Werbos, 1990). However, such recurrent networks exhibit complex dynamics, and are hard to train compared to feedforward networks. Furthermore, it has been shown that the error gradient needed to backpropagate errors vanishes in few times, which makes it difficult to maintain information over more than few time steps (Hochreiter and Schmidhuber, 1997). To alleviate this, Hochreiter and Schmidhuber (1997) proposed a new architecture called Long Short Term Memory (LSTM). The basic LSTM memory unit comprises a self-connected memory unit along with gate units that control the amount of information that flows in and out of the memory cell. This enables LSTM networks to learn data with extended context dependencies, but adds further complexity to the model, which limited its widespread use outside the neural network community. A more recent network architecture called Echo State Networks has been proposed by Jaeger (2003), which consists of a hidden layer with randomly sparse connections. Finally, Self Organizing Maps (SOM), (Kohonen, 1998)) are an unsupervised connectionist approach that use a pool of interconnected artificial units. All units are connected to the inputs, when a new input is presented, the units whose weights are most similar to the input is selected in a competitive fashion. The winning unit is then updated along with neighboring units. Since their inception, SOM have been widely used for visualization purposes, but have also been used to learn sequences of events by using hierarchies of SOM (Carpinteiro, 2000).

The review on causal models for sequence prediction we have proposed here is not exhaustive. Here we have focused on defining a basic set of tools that can be used when dealing with musical sequence modelling. We will now present their applications to music modelling.

3.2.5 Applications to music modelling

Markov-chain techniques have been considered in musical applications from machine improvisation (Lartillot et al., 2001) to cognitive modelling of music perception (Ferrand et al., 2002). Lartillot et al. (2001) compares Incremental Parsing and Prediction Suffix Trees to represent melodies and model musical style. Pearce and Wiggins (2004) study and evaluate statistical models for capturing the structure of monophonic melodies using 8 melodic datasets. Extensions of N-gram techniques such as Predictive Partial Match are compared. Other approaches make use of Markovian modelling to learn the structure of musical sequences in an interaction setting, the best known being the Continuator (Pachet, 2003) and OMax (Assayag and Dubnov, 2004). This latter system uses the oracle factor (Allauzen et al., 1999) to create a hierarchical representation of musical sequences that can be applied to MIDI signals or to melodies in audio signals (Assayag et al., 2006). Paiement (2008) investigates the use of graphical models to encode the dependencies of symbolic music sequences. Paiement proposes approaches to integrate musical knowledge such as musical meter and tonality in the learning process and shows that models that take into account this knowledge can perform better when encoding musical sequences. Additionally, methods based on evolutionary computation that rely less directly on markov-chains have been proposed to model musical style, with applications to improvisation (Homer and Goldberg, 1991; Biles, 1994; Burton and Vladimirova, 1999), or performance modelling (Ramirez and Hazan, 2005; Hazan et al., 2006).

Connectionist approaches have been long investigated for building models of music perception and performance. We refer to Todd and Loy (1991); Griffith and Todd (1999) for an overview. First, simple models have been proposed to build computational representations of musical sequence in a distributed way. Bharucha and Todd (1989) proposed a connectionist model of tonal representation. A three layer network was used, to represent relations between notes (first layer), chords (second layer) and keys (last layer). Incorporating tonal knowledge in the network was made possible by creating connections from one layer to another. All notes belonging to a specific chord were connected. Similarly, all chords belonging to a key were connected. This allowed the process incoming notes in a distributed way, by propagating the units activation among the three layers. However, the model proposed by Bharucha is hardwired, and doesn't learn tonal relationships through exposure. Tillmann et al. (2000) proposed a model based on two self-organizing map layers that was able to learn these relationships in an unsupervised way, by exposure to musical sequences. The connections obtained after training were comparable to the hardwired connections in (Bharucha and Todd, 1989). Other works have emphasized the sequential dependencies between musical events, without explicitly defining a hierarchical network architecture. In this context, recurrent network models have been considered because of their ability to encode sequences. Mozer (1994) used a model close to Elman's SRN to learn a variety of musical tasks, investigating either artificial melodies or Bach excerpts. Mozer compared

different representations of notes and chords and showed that an implicit hierarchical representation of musical events was maintained by the networks. The LSTM network architecture has been used more recently to learn the structure of blues excerpts (Eck and Schmidhuber, 2002) and generate blues improvisations, or to create models of jazz performances (Franklin, 2004). Other works aim at studying whether artificial neural networks can provide an account of musical learning as observed in human subjects. Kuhn and Dienes (2008) investigated whether time delay feedforward neural networks and simple recurrent networks could learn a non local music construction rule using a bi-conditional grammar and compared the results to those of a listening experiment.

3.3 Concluding remarks

Overall, this review shows that a considerable amount of research has been carried on, on the one hand to build and compare models of sequential learning, and on the other hand to apply them to musical tasks. Altough in most cases these works assume a fixed symbolic representation of low level musical events (e.g. tones, chords), some of them have shown that higher-level representations could be derived by exposure to musical material (Mozer, 1994; Tillmann et al., 2000). Also, some works have shown that it was possible to design music generation systems that could interact with users (Pachet, 2003; Assayag and Dubnov, 2004; Assayag et al., 2006), unveiling the applications of music modelling through sequence learning. However, fewer works have focused on investigating sequence learning from a music perception viewpoint, by attempting to compare the model behavior with experimental data on human subjects (Tillmann et al., 2000; Kuhn and Dienes, 2008). Even in this case, past research often omits considering alternative representations for musical sequences. In the next chapter, we will introduce an evaluation use case, in which we study how connectionist prediction models such as TDNN and SRN can provide an account of the behavioral results obtained by Saffran et al. (1999) by comparing a set of representations of tone sequences and will study how these alternative symbolic representations affect the simulation results. Additionally, it is worth noting here that the research presented above almost exclusively focuses on western tone sequences encoded in a symbolic way. The main principles enounced here (on-line learning, prediction, emergence of representations) may as well apply in a more general framework of music listening. In Chapter 6 we will introduce a model in which musical events may be described from audio signals in terms of spectro-temporal properties, timing and rhythm, thus allowing a mid-level approach to music modelling and expectation.

CHAPTER 4

A modelling use case: statistical learning of tone sequences

4.1 Chapter Summary

We aim at simulating the findings obtained by Saffran et al. (1999) that were described in Section 2.3. These simulations will provide us a starting point to validate models of musical sequence learning. We propose a validation loop that follows the experimental setup that was used with human subjects, in order to characterize the networks' accuracy to learn the statistical regularities of tone sequences. Tone-sequence encodings based on pitch class, pitch class intervals and melodic contour are considered and compared. The experimental setup is extended by introducing a pre-exposure forced-choice task, which makes it possible to detect an initial bias in the model population prior to exposure. Two distinct models, a Simple Recurrent Network (SRN) and a Feedforward Neural Network (FNN) with a time window of one event lead to similar results. We obtain the most consistent learning behavior using an encoding based on Pitch Classes, which is not a relative representation. More importantly, our simulations highlight the impact of tone sequence encoding in both initial model bias and post-exposure discrimination accuracy. The representation of symbolic information should not be determined *a priori* when simulating sequential learning experiments. Rather, the choice of the representation should be supported by physiological data or be compared and validated experimentally.

4.2 Motivation

The model we aim to develop in this thesis should take as input a musical stream and form specific expectations regarding the future sequence events, based on the sequence listened so far, and the internal representations developed by learning. Our approach constrains the model to be informed of music cognition findings as observed in psychological experiments. In our view, a main concern lies in finding specific musical and cognitive tasks that a model should perform to be validated. In this perspective, we exploit the idea that statistical environmental regularities are fundamental for perception and knowledge acquisition. Statistical regularities in the environment influence the processing of information in the brain, such as learning, memory and inductive inference. So far, humans' exploitation of these statistical regularities in cognitive mechanisms has been subject of study by cognitive sciences (Barlow, 2001). Learning, this way, could be seen as the internalization of environmental regularities, as a counterpart to Gestalt laws (proximity, common fate, good continuation), that describe bottom-up, hardwired processes. Learners take advantage of statistical information of syllable sequences such as the distribution of patterns of sounds to discover word boundaries.

As presented in Section 2.3, several works have devised an experimental protocol for assessing how humans learn regularities in acoustic sequences, made of either tones, phonemes, or timbres.

While these experiments can be seen as a means to validate models of expectation, we also suggest that computational simulations may be used to inspect further and eventually validate the experimental protocols themselves. As a starting point, we aim at simulating the experiment presented by Saffran et al. (1999).

4.3 Simulation setup

In this section, we provide details about our experimental setup, the alternatives we use in order to encode tone sequences, and our ANN models settings. First, we present in Figure 1 an overview of the experimental setting used in both original experiment and our simulation.

4.3.1 Tone sequence encoding

- Pitch Class (PC): each tone is encoded using a pitch class representation: we use 12 input units, for representing a given pitch we set the activation of one unit to one while the others are set to zero.
- Pitch Class Intervals (PCI): Each interval from one tone to the next one is encoded using a pitch class representation: we use 25 input units, for representing a given interval we set the activation of one unit to one while the others are set to zero. The 25 units allow intervals ranging from -12 to +12 semitones.
- Melodic Contour (C): Each interval from one tone to the next one is encoded using a contour representation: we use three input units, for

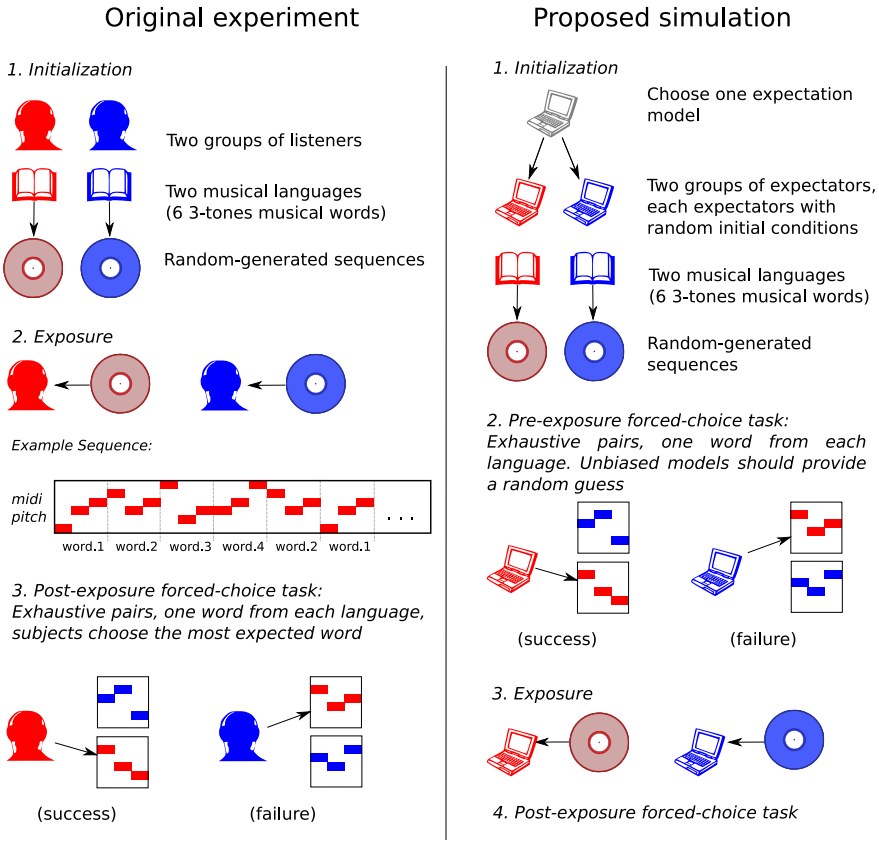


Figure 4.1: Overview of the experimental setup. left: original experiment from (Saffran et al., 1999), right: simulation.

representing a given interval we set the activation of one unit to one while the others are set to zero. The three units allow to represent the contours *down*, *same* and *up*.

4.3.2 ANN settings

ANN are usually trained in several passes, called epochs. Then a test phase, in which no weight update takes place, is subsequently performed. Here, by analogy with the approach of Kuhn and Dienes (2008), we make no distinction between training and test mode. At each time step, even during the forced-choice task, the network weights are updated to reduce the mismatch between their expectation and the next note event. The number of epochs is set to 1, because we want to reproduce a psychological experiment in which the subjects attend the sequence of stimuli only once. We use an FNN with a time window of one event, that is, the FNN network has only

access to the current event when predicting the next one. For both FNN and SRN, the detail of the parameters we explored is given below.

Exploration of the model parameters For our experiments, we used a set of parameters for defining and training the SRN. These parameters are learning rate, momentum, and number of hidden (and context) nodes. While the learning rate and momentum provide a way of controlling the weights search, the number of hidden and context units controls the complexity of the model that is learned. As a comparison with Kuhn and Dienes (2008), we do not allow a very large number of hidden and context units, for instance 60 or 120. This is because we believe that the task addressed by Saffran et al. (1999) involves smaller time dependencies than the bi-conditional learning task addressed by Kuhn and Dienes (2008). Thus, we use a smaller number of hidden units. We summarize the set of possible parameters in Table 4.1.

Table 4.1: Parameter Set for the SRN

Parameters	Values
learning rate	0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9
hidden units	2, 5, 10, 15, 30

4.3.3 Simulating the forced-choice task

In order to model the forced-choice task we compare, for each word tone or interval (depending on the selected coding schema), the model predictions with the actual next tone or interval. The word from which the lowest mismatch is observed is selected as the chosen word. Figure 2 shows how the forced-choice task is simulated for either interval-based encodings or tone-based encodings.

4.3.4 Experimental loop

We run our simulations using the following general loop.

1. Create, for each language, a random sequence of tones by concatenating words from the corresponding language. Following Saffran et al. (1999), we create, for each language, 6 blocks of 18 words each by randomly picking words from this language. Words never appear twice in a row. Then, the blocks are concatenated randomly to form sequences equivalent to a 21 minutes auditory stream.
2. Create two network instances for simulating an individual from Group 1 and another from Group 2. Both networks have initially random weights and activations
3. For each network, perform the forced-choice categorization task on all possible combinations of L1-L2 words. Store the recognition accuracy

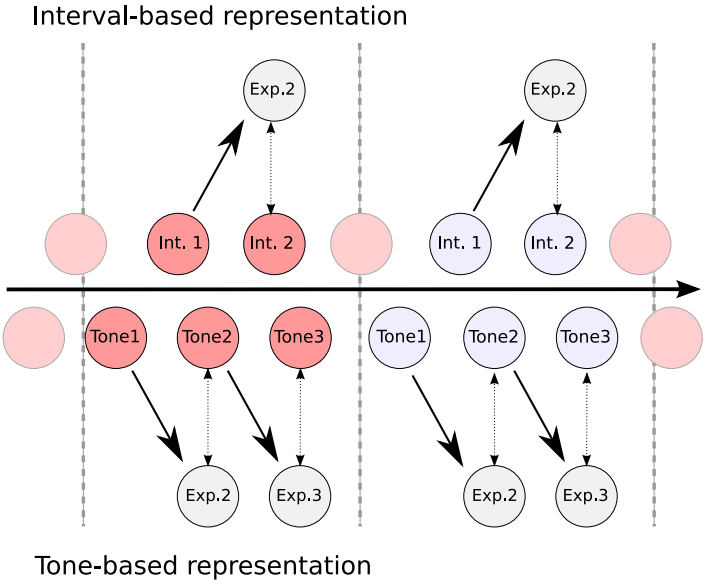


Figure 4.2: Forced-choice task simulation. The horizontal axis indicates time. Bottom: tone-based encoding, circles represent successive tones. Top: interval-based encoding, circles represent successive intervals. Plain diagonal arrows show models producing expectations of the next event. Only the events which are involved in the mismatch computation are labelled. Vertical bidirectional arrows show from which prediction the mismatch is measured. Gray vertical lines show word boundaries.

before exposure. During this task, the order of presentation of each stimulus pair is random. Moreover, when each pair is presented, the words from the two languages are presented in random order.

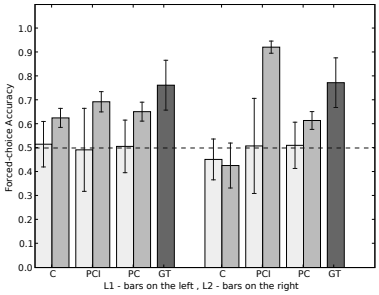
4. Present to each network its corresponding sequence.

5. For each network, perform again the forced-choice categorization task. The settings are similar to those presented in step 3. Store the recognition accuracy after exposure.

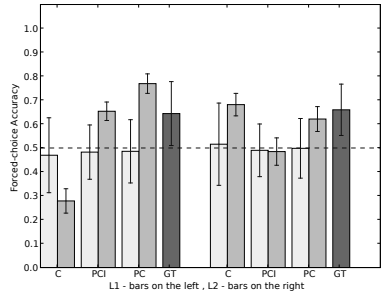
We repeat this loop 100 times for each experiment in order to extract a recognition accuracy score for each instance of the network.

4.4 Results and discussion

In Figures 4.3 and 4.4 we respectively show the results of our experiments involving the SRN and FNN models. For both models, the best results were obtained using the following parameters: 2 hidden units and a learning rate of 0.01.

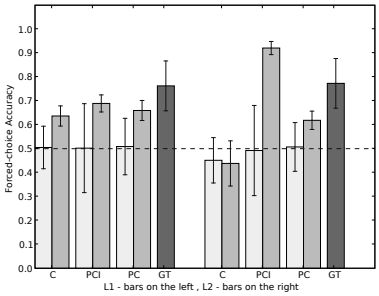


SRN model, Experiment 1

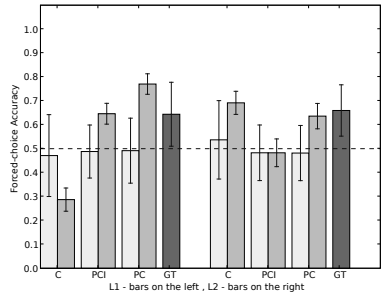


SRN model, Experiment 2

Figure 4.3: Forced-choice accuracy obtained with SRN predictor for distinct tone sequence encodings, compared with the subjects' response in (Saffran et al., 1999). The simulation of both Experiment 1 (left: words versus non-words) and Experiment 2 (right: words versus part-words). For each experiments and model, the results are shown for Language L1 on the left and for Language L2 on the right. Contour encoding is denoted C, Pitch Class Interval encoding is denoted PCI, and pitch class encoding is denoted PC. For each language, the right-most bar shows the ground truth post-exposure accuracy obtained by Saffran et al. (1999), denoted GT. The pre-exposure (light bars) and post-exposure (medium dark bars) mean scores are plotted, along with their standard deviation over the 100 runs. The horizontal dashed line indicates the 50% baseline.



FNN model, Experiment 1



FNN model, Experiment 2

Figure 4.4: Forced-choice accuracy obtained with FNN predictor for distinct tone sequence encodings, compared with the ground truth. For the details we refer to Figure 4.3.

4.4.1 Acquisition of statistical regularities

Inspecting the post-exposure results reveals distinct outcomes depending on the tone sequence encoding used. We analyzed the results obtained using independent samples t-tests when comparing pre and post-exposure scores, and used one-sample t-tests when comparing scores to the 50% baseline . Using the Pitch Class Interval representation, the post-exposure recognition scores are higher than the baseline for Experiment 1 for both languages ($p < 0.001$). However, both SRN and FNN models fail in reproducing the results of the second experiment, because the Language L2 post-exposure score is lower than the baseline.

The Contour-based representation can not account for the results of Experiment 1, because models exposed to Language L2 exhibit a post-exposure accuracy which is lower than the baseline ($p < 0.05$). Experiment 2 is not reproduced either: in this case the model population exposed to the Language L1 exhibits a strong negative post-exposure bias towards this language ($p < 0.001$). The fact that Experiment 2 involves the comparison of words versus part-words explains well the failure in obtaining a good fit using a melodic contour representation: indeed, the words to be discriminated during the forced-choice task are very similar when projected into a contour representation.

Overall, the most consistent improvement of the post-exposure forced-choice task accuracy for all experiments and languages ($p < 0.05$ in all cases) is obtained using a Pitch Class representation, that is, a representation where pitch is not defined with intervals. However, we were not able to reproduce the fact than Experiment 1, because it involves a comparison of words versus $n$on-words, led to a higher discrimination accuracy than Experiment 2. In our simulations, the average post-exposure accuracy for Experiment 1 is 65% for Language L1 and 61% for Language L2. For Experiment 2, the average post exposure accuracy is 77% for Language L1 and 63% for Language L2.

4.4.2 Influence of used architecture

The first observation to be made concerns the similarity between the results obtained in Figure 4.3 using the SRN model, and in 4.4 using the FNN model, independently of the encoding used. Our results suggest that the SRN model can not take advantage of a longer context when providing a prediction, which may confirm that the task presented in Saffran et al. (1999) can only be solved by means of computing transition probabilities between successive events.

We have proposed in this paper an attempt towards modelling the acquisition of statistical regularities in tone sequences. We have used two Artificial Neural Network architectures to simulate the general learning trend observed by Saffran et al. (1999). Our results show that the choice of the Artificial Neural Network architecture has little effect on the post exposure accuracy, which suggests that an extended temporal context is not necessary to model this task.

4.4.3 Influence of used representation

We have extended the original experiment with a pre-exposure forced-choice task and observed the outcome of this task with both simulations and a behavioral experiment. We have found that a bias towards a given language can appear, which may depend on the tone sequence representation used. This suggests that further studies aimed at investigating tone sequence learning should take into account different representations of the tone sequences.

The simulations based on interval representations such as Pitch Class Interval or Contour did not consistently account for the experimental results. That is, representations where the information between tones is relative , enabling shift invariance for instance, do not enable to reproduce the experimental results. However, using a tone sequence encoding based on Pitch Class, we observe, for all experiments and languages, an increase of the categorization accuracy of words versus non-words and words versus part-words in a population of prediction models after they have been exposed to tone sequences containing statistical regularities.

A note on scalability In the simulations presented in this chapter, we have proposed several alternatives to represent tone sequences. Our model provides a prediction of the next tone or interval based on the observation of a limited number of past events. We are interested in scaling our system in order to process more complex representations of musical events, over an increased temporal context. That is, we aim at working with a greater set of possible events, while being able to detect patterns of successive events, even if they are separated by more than a few time steps. In this context, it becomes more difficult to train the ANN architectures we have presented here in an online learning and prediction setting, where only one training epoch is allowed. This is the reason why we will use an alternative learning and prediction model when extending our system in Chapter 6.

4.4.4 Concluding remarks

As a summary of this chapter, we have presented a setup for simulating the forced-choice task experiments reported by Saffran et al. (1999) [1]. To achieve this, we have proposed a set of encodings based on absolute pitch values, intervals, and contours for representing the tone sequences used in the experiment, and have used two neural network architectures to implement the learning process that takes place with human subjects. We have found that the choice of the encoding has a major impact than the network architecture. Contrastingly, as stated in previous chapter, most of the works focusing on models of implicit learning for music stimuli use a unique representation of tone sequences when comparing learning models. Therefore, representation is an important aspect even when using a symbolic representation (Mozer, 1994; Conklin and Anagnostopoulou, 2001; Paiement, 2008).

1. The source code of the simulation framework is available `http://emcap.iua.upf.es/downloads/content_final/statistical_learning_experimenter_package.html`

In next chapter, we will present approaches to represent musical aspects in audio signals. Then, we will consider a representation of acoustic events through time that is not specific to melody in Chapter 6 and will take advantage of the conclusions obtained here to propose a set of representation alternatives. This way, we will be able to study the impact of the representation in the structure learning and expectation processes.

CHAPTER 5

Representation and Expectation in Music: a Mid-level approach

5.1 Chapter summary

In Chapter 3 we have introduced models of sequential learning and their applications to music modelling. In most cases, these models were applied to fixed, symbolic representations of musical events. In Chapter 4, we have compared the use of different symbolic representations in a sequential learning simulation. As our aim is to create a model of listening that can be applied to a range of musical audio signals, we need to define an interface between symbolic sequence modelling and audio representation approaches. In this chapter, we review approaches to automatic extraction of musical information from musical audio signals that can be used for further sequential processing. Therefore, our system will create symbols that correspond to musical objects present in a given audio stream, which makes our approach mid-level. Our representation must describe events at a higher level than the audio signal itself or quantities directly derived from its time-frequency analysis, but should at the same time be less abstract than a score because we want it to distinguish objects with different spectro-temporal properties and time locations.

First, we review transcription systems that identify musical events from their spectro-temporal properties and their location in time. Then, we show how audio analysis techniques that involve some predictive behavior can be regarded as simple expectation systems. Finally, we introduce prediction-driven systems and information-theoretic approaches that have been applied to audio signals. Based on this review, we will present in next chapter a model that integrates both representation and expectation processes.

5.2 Automatic description of signals with attacks, beats and timbre categories

In the last two decades, there has been a substantial amount of work aimed at providing a mid-level description of musical audio. Here, we narrow our presentation to works aimed at producing a mid-level description of musical signals in terms of timing and acoustic properties.

5.2.1 Time description: *when*

First, the incoming audio stream can be described in terms of onsets, that is, the beginning of a musical note or sound. Onset detection is commonly performed in two steps: first the incoming signal is transformed into a low-dimensional signal called the detection function. This can be done, for instance, using a bank of temporal filters derived from psychoacoustical knowledge (Klapuri, 1999), deriving a detection function from spectral domain features (Brossier et al., 2004; Collins, 2004; Duxbury et al., 2003b) or using supervised learning approaches (Lacoste and Eck, 2007). From this, a peak-picking component takes the detection function as inputs and returns onset times. The musical stream can also be described in terms of beats which characterize the stream periodicity even in the absence of clear perceptual attacks. Depending of the approach used, the output of such analysis can be the average periodicity, the position of the beats, or a location over a finer grained metrical grid. This approach is called beat-tracking or meter estimation, and is presented in Section 5.3.2.

5.2.2 Instrument and melody description: *what*

Apart from the time dimension, another mid-level description we are interested in is the nature of the sounds whose attack is detected. Applied to a melody, note height would correspond to pitch detection. Unpitched sounds may be identified using timbre categorization techniques. Both timbre and pitch properties may be combined to describe instrument notes in a precise way. The detection of unpitched sounds such as drum sounds and percussive sounds in general has been considered in a variety of MIR systems. Classification of unpitched sounds has been addressed by Gouyon et al. (2000); Herrera et al. (2002) and was followed by works focusing on the transcription of percussive excerpts such as drums (Gillet and Richard, 2004; Tanghe et al., 2005a; Yoshii et al., 2005) or beat-box (Kapur et al., 2004; Hazan, 2005c). While these approaches are essentially supervised, because they assign a label from a predefined set to the sounds they process, other works have proposed unsupervised approaches to categorization of sounds. (Wang et al., 2003; Paulus and Klapuri, 2003; Schwarz, 2004; Jehan, 2005; Hazan, 2005a), assuming a fixed number of timbre categories, or by considering hierarchical approaches Schwarz (2004); Jehan (2005). Indeed, unsupervised categorization requires a suitable distance measure between sounds to categorize. Casey (2002); Cano (2006); Pampalk et al. (2008) proposed approaches focused on similarity measures between sounds. For some of the

works presented here (Tanghe et al., 2005a; Hazan, 2005c), causal versions have been presented, enabling real time implementations.

5.2.3 Combining timbre and temporal Information

Paulus and Klapuri (2003) proposed an unsupervised transcription system for percussive events that uses both the timbre characteristics of strokes and their temporal features, derived from onset detection and meter estimation processes. The first stage on the system groups attacks depending of their timbre characteristics. After this stage, each attack is associated to a specific group, called timbre cluster, in an unsupervised way. Subsequently, timbre clusters are mapped to predefined drum labels by considering the rhythmic features of events within each cluster.

On a broader time scale, systems have been proposed to perform automatic segmentation of musical pieces, and are called structural segmentation systems. Examples of structural segmentation systems include (Foote, 1999; Aucouturier and Sandler, 2001; Peeters et al., 2002; Goto, 2003; Ong, 2006). The general idea lies in computing a self-similarity matrix over a description of the musical signal. Features include timbre (e.g. MFCC) or chroma features. The computation of a self similarity matrix involved in the segmentation makes it an inherently batch process.

We have presented an overview of automatic systems that aim to extract musical structures from audio signals. Those structures include events onsets, periodicity and metrical structure, and events categories based on spectro-temporal properties. Methods for evaluating the proposed systems have been proposed, and often require a comparison with data annotated by experts, known as ground truth. For each specific discipline (e.g. onset detection) evaluations of systems from several research teams using a common set ground truth are organized at MIREX [1]. Overall, the MIREX evaluations show that even if the accuracy of such systems increases over years, those systems are still error-prone and the issue of automatic description of musical audio can not be considered as a solved problem for the time being. Consequently, expectation models applied to audio signals need to take into account the errors introduced by automatic description stages.

5.3 Prediction in existing models of audio analysis

Most of the representation systems introduced above do not incorporate prediction in their processing chain. It is worth noting that the notion of signal prediction has long been considered in the audio signal processing field, at different time scales and levels of abstraction.

First, low-level techniques perform signal prediction at the sample level or over a short period of time. Then, beat-tracking techniques are based on a higher-level description of the musical signal because they extract information about its temporal structure, and some of those models can be con-

1. http://www.music-ir.org/mirex/

sidered as time expectation systems. Finally, a system of computational auditory scene analysis driven by prediction has been proposed by Ellis (1996).

5.3.1 Audio signal prediction at the sample level

One application of audio signal processing lies in describing the musical signal in a more compact form, thus allowing to discard redundant signal data and reducing transmission costs. One of the techniques that has been designed to perform audio signal analysis and compression is Linear Prediction Coding (LPC, (Makhoul, 1975)). LPC was originally developed to perform speech signals coding, and has been widely used since then in a variety of applications. The basic idea behind LPC is that speech signals can be described as the result of a simple model of the vocal tract, which contains a source and a filter. To produce vowels, the source uses a sequence of pulses (which originate from the glottis in the vocal tract). To produce consonants and unvoiced sounds, the source uses a noise-like signal. Then, the source signal is filtered by applying a filter response that is characteristic of a given phoneme. In the case of speech, 4 formants are usually used to define a phoneme response. Using the source and filter parameters, it is possible to re-synthesize the signal. Consequently, speech signal can be described in a compressed form using the source-filter parameters and its residual, which corresponds to the difference between original and the re-synthesized signal. LPC performs prediction at the sample level to compute the linear prediction coefficients for a given signal. Further investigations have proposed predictive models similar to LPC that incorporate psychoacoustic knowledge such as Perceptual Linear Predictive analysis (PLP) (Hermansky, 1990)) and models that focus on approximating the spectral changes rather than the spectrum itself such as RASTA (Hermansky and Morgan, 1994).

Another example of prediction-driven processing is the computation of a specific onset detection function, know as complex domain (Duxbury et al., 2003b). The complex domain detection function is computed on a spectrogram by using both magnitude and phase information. For stationary signals, we assume that each spectrogram bin evolves linearly from one frame to the next. It is then possible, for successive frames, to compute an estimate of each incoming bin based on its last values. From this, the error between estimated bins and actual bins is computed, resulting in the complex domain detection function.

Overall, the techniques presented here achieve a form of crude prediction on a low level basis, over a short time period. However, due to the short time scale of the predictions made, these methods can hardly be considered as models of of music perception. Following, we consider methods that analyze the periodicity of musical audio signals over a larger time scale: musical beats.

5.3.2 Beat-tracking as Expectation in Time

Beat tracking refers to the process of tracking periodic beat pulses in audio signals. These pulses correspond to the signal tactus and are often

easily estimated by listeners, even if they are non-musicians. From this, it is possible to perform a segmentation of the audio signal or extract its periodicity, for instance by computing the average Beats per Minute (BPM). Beat tracking algorithms have been proposed in the last three decades, using a variety of approaches, and were applied to either symbolic, MIDI or audio data. Some approaches attempted only to compute the average periodicity of a musical signal without identifying pulses, which is also referred to as tempo estimation. Other approaches can additionally identify pulses (Goto and Muraoka, 1995; Davies and Plumbley, 2005) or extract the finer-grained metrical structure of musical signals (Dixon, 1997; Cemgil et al., 2002).

The techniques used for performing beat-tracking range from rule-based models (Povel and Essens, 1985), autocorrelation methods (Davies and Plumbley, 2005; Brossier, 2006), oscillating filters (Scheirer, 1998), histograming methods (Gouyon and Herrera, 2003), multi-agent systems (Goto and Muraoka, 1995; Goto, 2001), probabilistic approaches (Cemgil et al., 2002), and multi-resolution analysis (Smith, 1996). We refer to Hainsworth (2006) for a detailed review of these approaches.

Here, we focus on approaches that can process audio signals as they can been seen as computational models of temporal expectation in the signal domain. We need to make a distinction between models that process the whole signal to be analyzed in a batch setting, and causal models that aim at extracting the periodicity, pulse or metrical structure in the audio stream analyzed so far, and that may be able to locate *incoming* time regions where musical events may occur. As discussed by Hainsworth (2006), methods such as oscillating filters make causal processing straightforward, while other methods can not perform causal processing or need to be adapted to do so, for instance by performing batch processing over a sliding analysis window.

From an expectation modelling viewpoint, a causal beat tracking system may be seen as a temporal expectation system. At each point in time, the system has to keep track of the most recent pulses and has to predict, based on the pulse history and the current signal, where the subsequent pulse will occur. Causal beat tracking is focused on expecting when the next pulse is going to be perceived, without considering which acoustical cues will be associated to that pulse. That is, they aim to answer the question of when next events will be heard, and can possibly associate these events to a finer-grained metrical structure. To this extent, beat-tracking approaches may provide a useful basis in the context of expectation modelling, and can serve either as a input of an expectation system or be part of the expectation model. However, beat-tracking approaches do not aim at distinguishing acoustical cues in audio signal, and do not inform about *what* is going to be heard next. When developing our expectation model in Chapter 6, we will show how some components of the system are related to beat-tracking approaches.

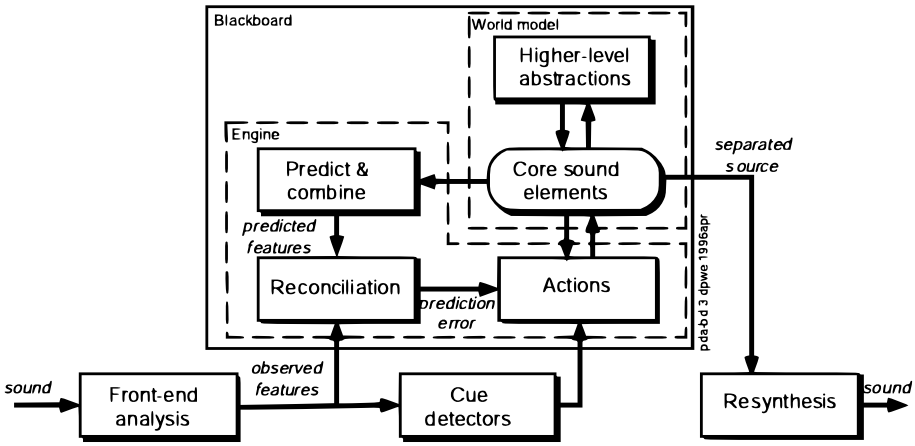


Figure 5.1: Block diagram of a prediction-driven computational auditory scene analysis system, from (Ellis, 1996).

5.3.3 Prediction-driven computational auditory scene analysis

The idea of building a system that can represent audio signals in a compact form to learn their structure and anticipate forthcoming events has been introduced by Ellis (1996). The author proposes a computational framework where the tasks of representing the acoustic stream and predicting its continuation are closely linked. This model is aimed to provide a more plausible alternative to standard auditory scene analysis models (Cooke, 1993; Brown, 1993). Those models are seen by the author as “data-driven" because they compute representations of the acoustic stream based on the low-level features only, without considering in which context these features are computed. Such data-driven approaches make it difficult to deal with ambiguous and noisy signals. Those ambiguities may be compensated in a model considering some knowledge about the signal structure. Furthermore, the author suggests that the temporal dynamics of the listening process requires to build an incremental model, which performs analysis in strictly advancing time steps, instead of of performing batch processing of waveforms. A block diagram of the presented model is shown in Figure 5.1.

The auditory front-end feeds the system core engine, which is is charge of *reconciliating* the low level observations with an ”internal model of the sound-producing entities in the environment“. That is, the engine has to adapt both the world model and the sensory input to find some degree of agreement between them. The world model is itself based on a representation of the signal, which has be chosen to provide a very general description, intended to deal with the broad auditory environment of human perception. Three acoustic units are used to describe the acoustic environment: noise units, tonal units and transients. The system does not take into account

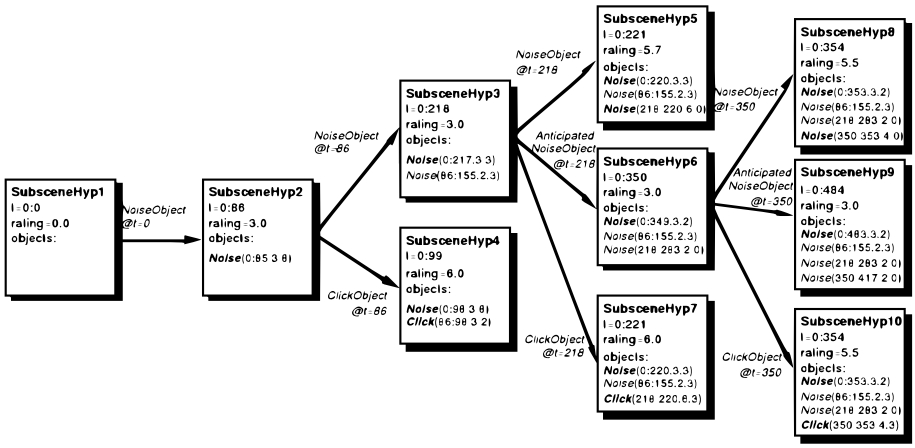


Figure 5.2: Diagram of the evolution of various alternative explanations from a simple example involving noise sounds, from (Ellis, 1996). From left to right: a root hypothesis concerning an event occurrence at a given time gives rise to a tree of possible observations concerning future auditory objects.

finer-grained details so that similar sounds with specific differences can be described the same way. This model makes use of a *blackboard architecture*, which is a functional model of memory and representation. The blackboard provides a computational framework to allow various competing hypotheses at different levels of abstraction to be considered which gives representational power to the system. We show in Figure 5.2 an example of competing representation that can emerge from Ellis system analyzing an audio excerpt of a simple experiment from Bregman.

The diagram shown in Figure 5.2 shows the various competing hypotheses that can emerge from the analysis of a simple excerpt. An issue raised by this type of model is the scalability of the system when considering a richer representation. If our aim is to model -even in rough form- the dynamics of musical excerpt, we will need to consider a richer representation of events for an extended temporal context. In this case, a blackboard approach would quickly face combinatorial explosion issues. Ellis' model is applied to some analysis tasks, which includes auditory material such as voices and environmental sounds. The system is able to re-synthesize its representation of auditory events, which allows validation by listeners. Ellis has formulated a key modelling principle: an incremental system with a prediction to perception feedback loop. However, we need to refine and adapt the model to accommodate the representation of higher-order structures that may be suitable to represent musical information.

5.4 Information-Theoretic Approaches

We finally review systems that use an information-theoretic approach to audio analysis and that have proposed a formalism to define prediction in audio signals. Abdallah (2002) considers redundancy reduction and unsupervised learning applied to musical and spoken audio (either waveform or spectral distribution), and reports a set of experiments aimed at defining a base representation of audio signals. Lewicki (2002) follows a similar approach to represent natural sounds. Abdallah also sketches how a musical system may exhibit degrees of *suprise* in a way rooted in perception and information theory. From an information-theoretic viewpoint, the musical signal can be viewed as an information source. This source is then transmitted to a listener (receiver) through a noisy information channel. As a consequence of this, the listener perceives both musical structure and noise associated to the transmission of the musical content. As such, information theoretic models include both the musical signal and the listener into the music perception model. This provides an alternative account to analysis of musical signals by considering the predictability of their structure from the listener's point of view instead of merely computing statistics on the musical signal itself as it is done in most works listed in Section 5.2 and more generally in MIR systems. According to Abdallah and Plumbley (2009), "the general thesis is that perceptible qualities and subjective states like uncertainty, surprise, complexity, tension, and interestingness are closely related to information-theoretic quantities like entropy, relative entropy, and mutual information." Pearce and Wiggins (2004) use *entropy* and *cross-entropy* to evaluate whether statistical models can learn symbolic monophonic melodies. The *entropy rate*, denoted $H(X|Z)$, reflects the instantaneous certainty of statistical models to characterize the current X observations given Z past observations , while *cross-entropy* applied to test melodies informs about the generalization accuracy of a learned statistical model.

Dubnov (2006) introduces a model of transmission of the musical signal over a time-channel. In this model, "The input to the channel is the history of the signal up to the current point in time, and the output is its next (present) sample." The channel receiver makes a prediction of the upcoming sample based on the history of samples it has received so far. Therefore, it is possible to compare the receiver prediction with the next sample to be transmitted. The *Information Rate* (IR) is a measure of how the information is transmitted from the source to the receiver in the time-channel. Dubnov derives the IR model from this measure, and applies it to scalar and multivariate musical signals. The unidimensional IR case is shown to be equivalent to using a descriptor known as Spectral Flatness (Jayant and Noll, 1984). In the multivariate case, the model can process musical signal represented with multi-dimensional audio descriptors. The model is extended in (Dubnov, 2008), where an additional recurrence analysis component enables to detect repetitions in blocks on a larger time scale. In this case, two parallel processes take place, one focusing on short-term frames of signal to compute a so-called data-IR, and the other focusing on long-term frames to compute the model-IR. Applications to audio signals such as natural sounds

and classical music are presented in (Dubnov, 2006, 2008). Cont (2008) discusses and extends the IR model in his thesis. Abdallah (2008) discusses the validity of Dubnov's IR measure and proposes further information measures in (Abdallah and Plumbley, 2009), where the authors propose to compute the *average predictive information rate*, noted $I(X, Y|Z)$ which may be seen as the average rate at which new information arrives about the present and future observations X and Y, given past observations Z. In an experiment using a simple Markov-chain model applied to two Philip Glass minimalistic music pieces, the authors show that these measures reveal the structure of the pieces in agreement with the judgment of a human expert listener. However, Abdallah's model was not tested on representations derived from audio signals.

5.5 Concluding remarks

Overall, information-theoretic approaches provide a convenient and elegant framework to formulate issues such as signal representation and expectation, and may be applied to different representations of musical signals, either in a symbolic form or as audio excerpts. Furthermore, the idea of deriving a useful representation of musical signals in a purely unsupervised way - instead of crafting specific representation algorithms for specific tasks - is appealing. However, more work is needed in this area to give researchers ways of extracting musical structure as those found in state of the art MIR systems. As an alternative to pure MIR-centered and Information-theoretic systems, we will propose in next chapter a hybrid approach. Some typical components from MIR systems will be used to provide a mid-level representation of audio events, those events will in turn be processed in an information-theoretic fashion to further organize them and enables their prediction.

CHAPTER 6

The What/When expectation model

6.1 Chapter summary

Our model of music expectation is aimed at combining audio signal representation and the generation of future occurrences in this representation within a single framework. A causal system to represent a stream of music into musical events, and to generate further expected events, is presented here. Starting from an auditory front-end which extracts timbral and temporal features such as onsets and beats, a categorization process (either supervised or unsupervised) builds and maintains a set of symbols aimed at representing musical stream events using both timbre and time descriptions. The time events are represented using inter-onset intervals relative to the beats. We propose and compare three ways of combining time and timbre dimensions regarding the prediction of the next event. Several alternatives to timbre and timing description, cluster estimation and assignments are considered.

These symbols are then processed by an expectation module using Predictive Partial Match, a multiscale technique based on N-grams. To characterize the ability of the system to generate an expectation that matches both ground truth and system transcription, we introduce several measures that take into account the uncertainty associated with the unsupervised encoding of the musical sequence. The system is evaluated using a database of annotated drum recordings. We compare three approaches to combine timing (*when*) and timbre (*what*) expectation. The applications of this models, including visualization and sonification of the expectation, are presented in Chapter 7.

6.2 Motivation

Our main focus lies in integrating a learning system which can constantly learn aspects of the structure of musical audio signals *while it listens to musical events*, in a way that is inspired by cognitive principles. We propose a causal and unsupervised system that learns the structure of an audio stream and predicts how it will continue. Our system uses concepts and approaches from a variety of research topics: automatic music transcription, unsupervised learning and model selection, and symbolic statistical learning. A mid-level representation of the signal is constructed as a discrete sequence of symbols representing time dependencies between events and timbre properties. A prediction of the subsequent symbols can then be provided, from which the system can then predict the nature and the timing of the next musical event. As such, the system prediction algorithm is symbolic, but the system is responsible of producing a symbolic representation of the musical stream in an unsupervised manner (i.e. the symbols are based on acoustic information).

From a general viewpoint, we assume a musical sequence as being a succession of musical events with intrinsic properties (e.g. pitch, loudness, timbre) which are heard at a given time. While the musical sequence is attended, timbre and timing patterns perceived so far (among other musical information that is not considered in this paper) can be used to provide a prediction of the next musical event to be heard. The prediction can be made over several dimensions, e.g. the timbre properties of the next event to be heard (*what*) or the temporal location of the next event (*when*). Distinct strategies may be used by listeners to combine predictions along these dimensions. Even tough in the case of timbre recent evidence points to the possibility that certain timbre dimensions (e.g., attack slope, spectral centre of gravity, etc.) could be processed separately (Caclin et al., 2006) we will consider timbre as a nonseparable dimension to be used for predicting events. From the physiological literature, it seems to be a separation between the circuits dealing with the detection of violations of musical expectancies or predictions (in the ventrolateral prefrontal cortex), and the circuits dealing with the processing of perceptual objects and their monitoring in working memory in the dorsolateral prefrontal cortex and posterior parietal cortex (Sridharan et al., 2007).

In addition, we may assume a functional and physiological separation between sequential processing (related to musical syntax and grammar) which could take place in Broca's area (Maess et al., 2001) and the processing of timing information, which could be happening at the right temporal auditory cortex and superior temporal gyrus (Peretz and Zatorre, 2005)). From these physiological considerations we could address the prediction of what and when dimensions as two independent processes which may be modeled with two independent predictors.

However, from the Auditory Scene Analysis (Bregman, 1990) point of view, sound events may be separated into different auditory objects (auditory stream segregation) or assigned to a single auditory object (auditory stream fusion). This depends on the intrinsic properties of the musical events

(e.g. timbre, pitch) and the timing of the events. If auditory stream integration takes place, each event may be described as how it differs from the preceding event in terms of timbre and timing. In this sense, the *what* and *when* dimensions might be merged into a unique dimension, which would be modeled with a unique next-event predictor. Finally, if auditory stream segregation takes place, a different representation of the events has to be considered. In a perceptual experiment (Creel et al., 2004), while the statistical regularities between nonadjacent tones were hardly learned when the musical sequence contained events which were similar in pitch or timbre, these regularities could be acquired when the temporally nonadjacent events differed in pitch range or timbre. Therefore, if a musical sequence can be perceptually segregated into separate timbre streams, the temporal dependencies between events might be computed (to a certain extent) from the separate timbre stream rather than from the temporally adjacent elements. This could be modeled by considering each segregated stream as a separate dimension which would be modeled with a stream-specific timing predictor. The system we present in this chapter enables us to implement these three strategies and compare them empirically using a database of annotated sequences of percussive events.

6.3 Overview of the system

The system has the following modules: *feature extraction*, *dimensionality reduction*, and *next event prediction*. These components, all of which run simultaneously when the system is following a musical stream, are shown in Figure 6.1. First, the *feature extraction* module is the audio front-end, which extracts timbre descriptors, onsets and beats from the incoming signal. This module is based on the Aubio library (Brossier, 2006). Each extracted hit is encoded in the *dimensionality reduction* module based on both time and timbre description, following an unsupervised scheme. Therefore, we obtain a symbolic representation of the incoming events, to be used by the *next event prediction* module.

6.4 Low and Mid-level Feature Extraction

6.4.1 Analysis Settings

The audio stream is analyzed using a window size of 1024 samples (23 ms using a sampling rate of 44 KHz) with 50 percent overlap. We apply a Hamming window before computing the Fast Fourier Transform (with the same window size) to perform spectral analysis.

6.4.2 Temporal detection

Onsets are extracted as events are presented. Optionally, beat locations are extracted, and combined with onsets to produce a tempo-independent timing characterization between successive events. We present the methods used for achieving both onset detection and beat tracking tasks.

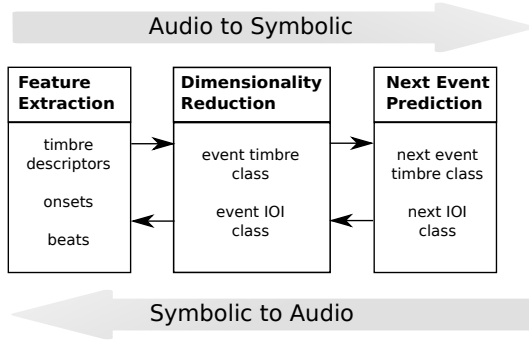


Figure 6.1: System diagram. Feedforward connections (left to right) create a stream of symbols to be learned. Feedback connections (right to left) enable symbolic predictions to be mapped back into absolute time. IOI refers to Inter-Onset Interval, as explained in the next section.

Onset detection We compare different onset detection techniques that are available in the Aubio library. The methods we compare are the following:

- High-Frequency Content (HFC, (Masri, 1996)), obtained by summing the linearly-weighted values of the spectral magnitudes
- Complex domain (Duxbury et al., 2003a), obtained by observing the fluctuation of the spectrum in the complex domain, thus taking advantage of both phase and magnitude.
- Dual: a hybrid function which combines the complex domain function with a function based on the KL-divergence.

Details and evaluation for these algorithms are available in (Brossier, 2006).

Beat tracking Beat tracking can be applied to provide time anchors from which it is possible to describe inter-onsets intervals in a *beat relative* fashion. This would enable the system to create a tempo-invariant representation of time. The tempo detection algorithm is based on (Davies et al., 2005). This algorithm is based on the autocorrelation of the onset detection function. A comb filter is applied to the resulting autocorrelation function, leading to an histogram of the period candidates. The histogram peak is then selected as the detected period, denoted $Period(t)$.

6.4.3 Inter Onset Intervals Characterization

We propose here alternatives to characterize timing relations between events. As an outcome of the temporal detection, the duration between successive events can be measured. These intervals can be seen as an absolute difference of onset times, or as beat-relative difference.

Inter Onset Interval If an onset occurs at time t_{curr}, and the previous onset has occurred at time t_{prev} we can derive the absolute Inter Onset Interval (IOI) as:

$$IOI(t) = t_{curr} - t_{prev} \tag{6.1}$$

Beat-Relative Inter Onset Interval Based on this, for each new event, the beat-relative inter-onset interval (BRIOI) is computed as follows:

$$BRIOI(t) = \frac{IOI(t)}{Period(t)} \tag{6.2}$$

where $IOI(t)$ refers to the inter-onset interval between the current event onset and the previous onset, and $Period(t)$ refers to the current extracted beat period.

Cluster-Wise Inter Onset Intervals Each incoming event belongs to a timbre class. In Section 6.5 we propose an unsupervised approach to assign a timbre symbol to incoming events. This makes it possible to compute IOI between events that belong to the same timbre category, instead of characterizing IOI between successive events of distinct timbre classes.

Beat Relative Cluster-Wise Inter Onset Intervals can be computed similarly to BRIOI.

6.4.4 Timbre Description

We use 13 Mel-Frequency Cepstrum Coefficients (Davis and Mermelstein, 1980). We have implemented the MFCC in the Aubio library (Brossier, 2006) following Slaney's MATLAB implementation, using 40 filters. We compute the median of each coefficient over a 100 ms window starting from the onset frame, in order to represent both attack and tail of the detected event. The value of 100 ms is chosen as a tradeoff allowing to either represent short, percussive sounds or longer sustained sounds.

6.5 Event quantization in the time and timbre space

In the dimensionality reduction and event quantization stage, our system assigns detected events to discrete time and timbre categories, thus creating a symbolic description of the audio stream. This is done by applying dimensionality reduction and clustering techniques, as shown below. Therefore, the output of this stage a set of cluster assignments corresponding to timbre and timing properties of the detected events. We will see in Section 6.6.2 alternatives to recombine these cluster assignments to obtain the desired symbolic representation.

6.5.1 Bootstrap step

Before starting to effectively encode and expect musical events, the system accumulates observations and therefore acts having a short-term memory buffer to gather statistics based on the incoming hits. During this accumulation period, the system does not provide any prediction regarding the future events. The processes involved here are (a) feature normalization, (b) Principal Component Analysis (PCA) for dimensionality reduction and (c) estimation of the number of clusters, for both timbre features and IOI. While (a) and (c) are always performed, (b) is only applied to the timbre features, and it is optional.

Bootstrap feature preprocessing

- Feature normalization: we normalize the accumulated timbre descriptors and IOI so that they have zero-mean and unit variance. The initial distribution of parameters is stored so that any normalized instance can be mapped back into its initial feature space.
- PCA: A PCA can be performed on the bootstrap normalized timbre features. In this case, instead of choosing the target dimensionality, we choose the desired amount of explained variance of the projected set compared to the original set.

The bootstrap information is stored as it enables to subsequently perform normalization and dimensionality reduction on new data (Figure 6.1, left to right connection), or to expand and apply inverse normalization to the projected data (Figure 6.1, right to left connection). Additionally, we also store the normalized and projected short-term history, which is used to estimate the number of clusters to work with. This step is presented in the next paragraph.

Evaluating the number of symbols The number of clusters to represent both IOI and timbre events influences the performance of the system and has to be chosen carefully during the bootstrap step. We perform first a cluster estimation using a grid of Gaussian Mixture Models with diagonal covariance matrix, trained with the Expectation-Maximization (EM) (Dempster et al., 1977) algorithm, following a voting procedure derived from (Cournapeau, 2006).

If our maximum number of clusters is fixed to M, we need to perform model fitting for all the cluster values from 1 to M and select the best model by taking into account both likelihood to the data and model complexity. Additionally, for each cluster number, we run R independent processes, which provides a better robustness to our estimation. Therefore, we create an estimation grid in which $R * M$, where M is the maximum number of clusters we allow, and R is the number of independent runs. Each column of the grid represents R models with an increasing number of clusters, from one to M. We train each grid model with EM, using 20 iterations. Once the grids are trained we proceed to the model selection step by computing information criteria as explained below.

Information criteria for model selection Each grid model can be described with the following parameters. First, the maximized likelihood, denoted by L, is a quantitative measure of how the trained model fits the data. The number of free parameters, K, measures how complex the model is. The number of samples N is the number of instances present in the short-term history (see previous paragraph). Different information criteria (IC) have been used in the model selection literature, which are described below. First, the Bayesian Information Criterion (BIC) (Schwarz, 1978) is defined as follows:

$$BIC = -2\ln(L) + K\ln(N) \tag{6.3}$$

The BIC strongly penalizes complex models. Models with few parameters and which maximize the data likelihood minimize the BIC. Akaike (1974) proposes another information criterion which penalizes less the model complexity, but does not take into account the amount of available data.

$$AIC = 2K - 2\ln(L). \tag{6.4}$$

In Section 6.7, we will compare the performance of the system when either BIC or AIC are used to determine the number of clusters in the data. To decide the final number of clusters we compute the median over the cross-runs:

$$K = \text{median}_{1\leq i\leq R}(\text{argmin}_{1\leq j\leq M}(IC)) \tag{6.5}$$

As an outcome, the estimated number of timbre and IOI clusters is noted K_{Timbre} and K_{IOI} respectively.

6.5.2 Running state

Once the clusters have been estimated for timbre features and BRIOI, we have to generate cluster assignments for incoming instances and update the clusters to take into account these new instances. To achieve this, we use an online K-means algorithm, with each cluster mean vector being initialized by the GMM model selected at the end of the bootstrap step. A cluster is assigned to each instance x following:

$$k_x = \text{argmin}_{1<k<K}||x - \mu_k|| \tag{6.6}$$

where μ_k is the k-th cluster mean. Then the mean of the assigned cluster is updated following:

$$\Delta\mu_{k_x} = \eta(x - \mu_{k_x}) \tag{6.7}$$

Here η is the *learning rate*, which controls how much each new instance influences the mean update of its assigned cluster. Values near zero have the effect of quantizing the incoming instance to the cluster mean, while values near one tend to shift the cluster mean towards the instance assigned to it. We have experimented with values comprised between 0.1 and 0.9, and have also implemented an optimal learning rate schedule, following (Bottou, 2004). In this latter case, the learning rate depends on how many instances a given cluster centroid represents.

Then the optimal learning rate can be computed as:

$$\eta_k^{opt} = \frac{1}{n_k} \tag{6.8}$$

where n_k represents the current number of data points in cluster k.

We illustrate the overall timbre encoding process in Figure 6.2 using a commercial drum'n bass pattern (Audio.1). The normalized distance involved in the cluster assignment step is obtained at the end of the bootstrap step. A principal component analysis has been used to project the internal timbre description into two dimensions.

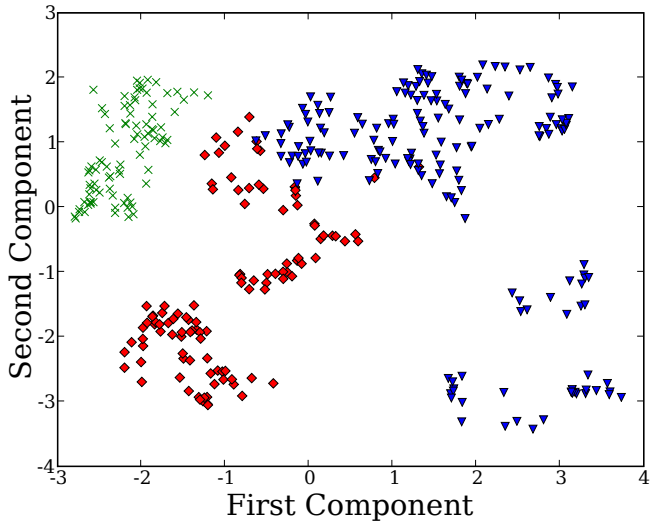


Figure 6.2: Timbre clusters assigned to each event after exposure to a commercial drum'n bass pattern (Audio.1). The timbre descriptors are MFCC. Crosses, squares and triangles represent points assigned to a specific timbre cluster.

Finally, we show in Figure 6.3, the BRIOI cluster assignments when processing the same excerpt. The figure shows the raw, unclustered BRIOI histogram (bottom), and the histogram of clustered BRIOI (top), i.e. in which each BRIOI event has been substituted by the current mean of its assigned cluster. We can see that the IOI clustering process performs a kind of "soft" quantization of the raw IOI values.

6.6 From representation to expectation

6.6.1 Multi-scale N-grams

The prediction module has to deduce the most likely future events based on the sequence observed so far. We treat the incoming encoded signal as

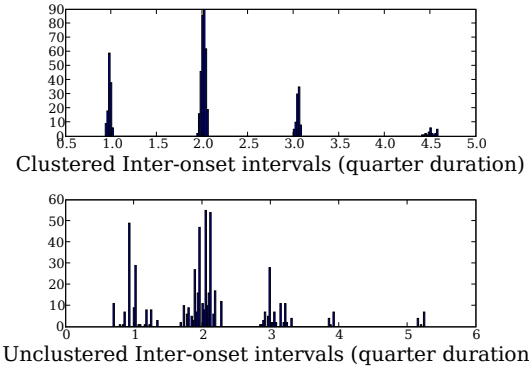


Figure 6.3: Unclustered and clustered BRIOI histograms after exposure to a commercial drum'n bass pattern.

a sequence of symbols and use a symbolic expectation algorithm. In this work, we use the Prediction by Partial Match (PPM) (Cleary and Witten, 1984) algorithm. PPM is a multiscale prediction technique based on N-grams, which has been applied to lossless compression and to the statistical modelling of symbolic pitch sequences (Pearce and Wiggins, 2004). In N-gram modelling, the probability distribution of the next symbol is computed based on the count of the sub-sequences preceding each possible symbol.

The probability distribution of the next symbol e_i, where $1 \leq i \leq K$, given the context sequence $e_{(i-n)+1}^{i-1}$ is:

$$p(e_i|e_{(i-n)+1}^{i-1}) = \begin{cases} \alpha(e_i|e_{(i-n)+1}^{i-1}) & \text{if } c(e_i|e_{(i-n)+1}^{i-1}) > 0 \\ \gamma(e_{(i-n)+1}^{i-1})p(e_i|e_{(i-n)+2}^{i-1}) & \text{if } c(e_i|e_{(i-n)+1}^{i-1}) = 0 \end{cases}$$

where $c(e_i|e_{(i-n)+1}^{i-1})$ is the number of counts of each symbol e_i following the subsequence $e_{(i-n)+1}^{i-1}$. The symbol counts, given each possible subsequence of size n, are stored in a transition table containing K^n rows and K columns. When a symbol has not appeared after $e_{(i-n)+1}^{i-1}$, the model performs a recursive backoff to a lower-order context. Here we use a PPM model with escape method C (Moffat, 1990) and update exclusion, which provides a reasonable tradeoff between accuracy and complexity. In addition, γ and α are defined as follows:

$$\gamma(e_i|e_{(i-n)+1}^{i-1}) = \frac{t(e_{(i-n)+1}^{i-1})}{\sum_K c(e_{(i-n)+1}^{i-1}) + t(e_{(i-n)+1}^{i-1})} \tag{6.9}$$

$$\alpha(e_i|e_{(i-n)+1}^{i-1}) = \frac{c(e_i|e_{(i-n)+1}^{i-1})}{\sum_K c(e|e_{(i-n)+1}^{i-1}) + t(e_{(i-n)+1}^{i-1})} \tag{6.10}$$

In Equations 6.10 and 6.9, the quantity $t(e_i^j)$ is the number of different symbols which have appeared in the subsequence e_i^j, $j > i$.

6.6.2 Combining timbre and time: expectation schemes

The PPM predictor presented above produces an expectation of the next symbol to be observed given the observed context. We are interested in providing two predictions, concerning the timbre category and IOI category of the future event to be perceived. We propose to compare three expectation schemes whose graphical models are illustrated in Figure 6.4.

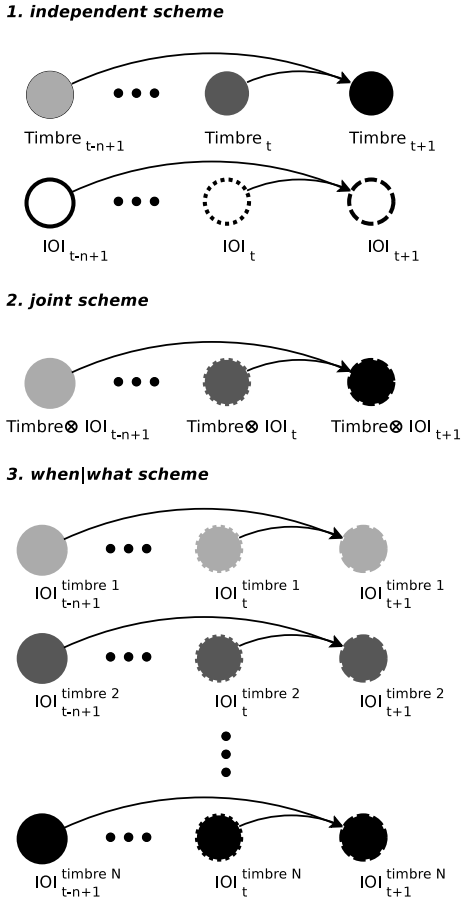


Figure 6.4: Graphical models of three schemes for combining what and when prediction.

Independent what/when prediction scheme Two independent PPM symbolic predictors are used. The random variables *Timbre* and *IOI* are thus considered independent.

Joint prediction scheme After the bootstrap step, the number of timbre and IOI symbols K_{Timbre} and K_{IOI} has been determined. From this, a new set of symbols denoted Timbre⊗IOI is created. This set contains exhaustive combinations of timbre and IOI symbols, therefore it has $K_{Timbre} * K_{IOI}$ elements. This approach can lead to high memory requirements if $K_{Timbre} * K_{IOI}$ becomes high. In this case, a unique PPM predictor is used to predict the symbol $Timbre \otimes IOI_{t+1}$.

When|what prediction scheme Each timbre cluster is associated with a specific cluster-wise IOI symbol predictor (see Section 6.4.3). This means that for each timbre cluster, there is a predictor which provides a guess of when an event belonging to the same timbre cluster will appear.

6.6.3 Scheme-dependent representation architecture

Depending of the expectation scheme used, the representation layer processes need to be adjusted to create a relevant set of timbre and/or time symbols. Here, we show the building blocks of the representation layer for the following schemes:

- Joint and independent scheme: both timbre and time symbols are created to either serve as input to two independent predictors (independent scheme) or be recombined into a set of combined timbre/time symbols (joint scheme) We show a block diagram of this configuration in Figure 6.5.
- When|what prediction scheme: IOI between events of the same timbre category are categorized by category-wise online K-means processes. Symbols are then given as input to a set of IOI symbol predictors, one for each category. The representation block diagram for this configuration is shown in Figure 6.6.

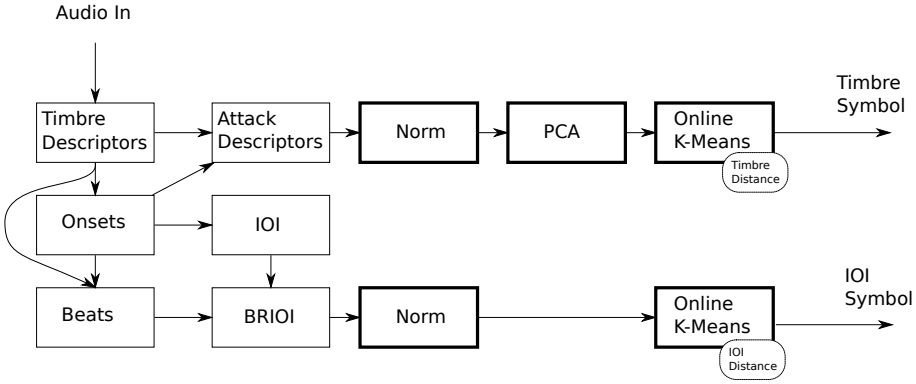


Figure 6.5: Block diagram of the representation layer used for independent and joint expectation schemes. Directed arrows represent the data processing flow, thick boxes represent processes that are initialized during the bootstrap step. Additionally, the distances involved in the categorization process are shown with rounded boxes.

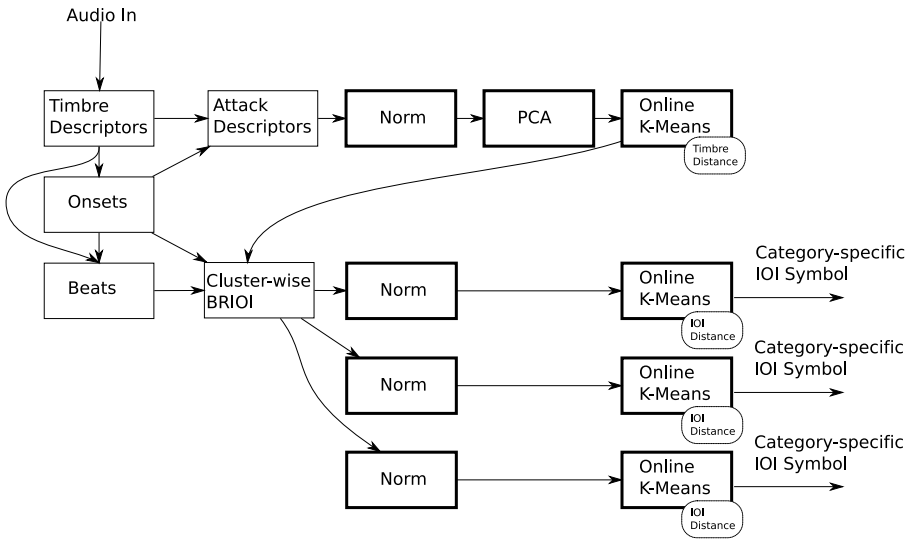


Figure 6.6: Block diagram of the representation layer used for when|what scheme. In this example, the number of timbre clusters is three, as result of the estimation in the bootstrap step. Directed arrows represent the data processing flow, thick boxes represent processes that are initialized during the bootstrap step. Additionally, the distances involved in the categorization process are shown with rounded boxes.

6.6.4 Unfolding time expectation

Based on the symbolic expectation generated, we can produce a timbre and BRIOI symbol expectation. From this, we apply an inverse normalization of the mean of the chosen BRIOI cluster and scale it to the current extracted tempo to obtain the absolute time position of the expected onset for the expected timbre cluster. Following we show in Figure 6.7 both transcription and expectation timelines obtained during exposure to a drum example. Here K_{Timbre} equals 3. While the beginning of the expectation timeline only contains events of timbre cluster #1 with random inter onset intervals, after a few seconds events from timbre cluster #3 start to be predicted following the transcription pattern. Then, the IOI pattern involving timbre cluster #1 events is learned, progressively followed by the timbre cluster #2 pattern.

6.7 System evaluation

In this section we present an evaluation of the system using a database of drum patterns. We first introduce a range of performance metrics for this

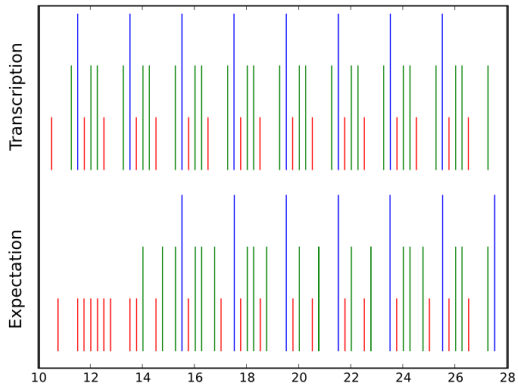


Figure 6.7: Comparison of transcription (top) and expectation (bottom) during exposure to an artificial drum pattern. Vertical colored lines indicate detected and expected events belonging to different timbre clusters. The vertical axis indicates time, after the bootstrap step has been performed.

task. We then present an experiment involving predictive learning of drum patterns.

6.7.1 Performance metrics

Our system produces an on-line transcription of the incoming audio stream and estimates, for each run, the optimal number of clusters to be used to produce a transcription. The transcription is used in turn to produce an expectation timeline which contains the same number of clusters than the transcription. Consequently, by using a database of annotated audio excerpts, several comparisons can be made. First, the transcription accuracy can be computed. Then, the transcription and the expectation produced by the system can be compared, without taking into account the annotations. To this respect, it would be desirable to know about violations of expectancies when human listeners learn a set specific musical patterns. Unfortunately such experiments have not yet been performed, and would require a careful choice of rhythms and acoustic stimuli. Finally the expectation timeline can be compared to the ground truth annotations. In this section, we present the performance metrics we use to achieve all these steps.

Comparison with the ground truth

Precision and recall applied to unsupervised transcription In the context of unsupervised transcription it might be useful to compare a sequence of events transcribed in an unsupervised way with a ground truth

annotation, where each ground truth event belongs to a fixed set of labels. If the sequence transcribed is labeled we can evaluate the analysis derived from the event detection and unsupervised clustering processes. We use a measure introduced in (Marxer et al., 2007) and (Marxer et al., 2008) that is designed to evaluate clustering when the mapping between the reference classes and estimated clusters is unknown. The confusion matrix is first constructed by using the onset matching technique presented by Brossier (2006) adapted to multiple classes of onsets. Let us consider C the number of ground truth classes and K the number of clusters. We write $n_{c,k}$ the number of co-occurrences of class c and cluster k, n_c the total number of occurrences of class c. Then we can express the precision and recall as:

$$P(c,k) = \begin{cases} 1 - \frac{\sum_{1 \leq i \leq C, i \neq c} n_{i,k}}{\sum_{1 \leq i \leq C, i \neq c} n_i} & \text{if } C > 1 \\ 1 & \text{otherwise.} \end{cases} \tag{6.11}$$

$$R(c,k) = \begin{cases} \frac{n_{c,k}-1}{n_c-1} & \text{if } n_c > 1 \\ 1 & \text{otherwise.} \end{cases} \tag{6.12}$$

The pairs of precision and recall of each cluster are integrated to achieve precision and recall measures per class. The integration is performed by doing a weighted average of the precision and recall values of the co-occurrences, among all occurrences of class c. The total precision and recall measures are the weighted sums of the per-class measures.

We will use P and R to evaluate the transcription and prediction accuracy of our system. When comparing the transcription timeline with the ground truth we will call the precision measure *Transcription Incremental Precision (TIP)* and the recall measure *Transcription Incremental Recall (TIR)*, and derive from these measures the *Transcription Incremental F-Measure (TIFM)*. When comparing the expectation timeline with the ground truth we will use the terms *Expectation Incremental Precision (EIP)* for P and *Expectation Incremental Recall (EIR)* for R, and from these measures we derive the *Expectation Incremental F-Measure (EIFM)*

Other useful metrics In addition to the measures defined above, we also consider a simple metric which compares the complexity of the system representation with the timbre complexity of the attended signal. This can be done straightforwardly by computing the class to cluster ratio as follows:

$$CCR = \frac{K_{Timbre}}{C_{Timbre}} \tag{6.13}$$

where C_{Timbre} is the number of different annotated timbre labels for this excerpt, and K_{Timbre} has been estimated at the end of the bootstrap step (see Equation 6.5).

Finally, it may happen, if the sensitivity of the system is low, that the bootstrap step leads to an estimate of timbre clusters equal to one. This is clearly not desirable in the context of evaluating a system which combines time and timbre dimension. For this reason, we introduce another statistic, which we name $P1$, and define as the percentage of runs in which the estimate of clusters led to one timbre cluster.

Comparing expectation and transcription

Comparing expectation and transcription timelines is easier than a comparison against the ground truth, mainly because both timelines share the same cluster representation. That is, the list of transcribed event onsets indexed by cluster n can be compared directly to the the list of expected event onsets indexed by cluster n by using onset detection related measures. We could compute the average F-measure by comparing both transcribed and expected timbre cluster onset times for each of the timbre clusters. However, such metrics cannot be considered here because of the variability of the unsupervised encoding we use. That is, the encoder provides a number of timbre clusters that is an approximation of the number of instrument or acoustic categories. For instance, the encoder can return an estimate of five timbre clusters for an excerpt containing three instruments. Consequently, during the running state, few of these clusters are indeed used, because a vast majority of the incoming instances are assigned to a subset of the estimated clusters (in the example, three of the five estimated clusters). As a result, the underrepresented clusters are likely to return very low F-measures, which may in turn affect the average computed F-measure.

In (Hazan et al., 2007) we have proposed to use the weighted average F-measure, which is defined as follows:

$$WFM = \sum_{i=1}^{K_t} w_i F_i \tag{6.14}$$

where K_t is the number of timbre clusters, each w_i is obtained by dividing the number of onsets assigned to cluster i by the total number of onsets, and F_i is the standard F-measure between onsets assigned to cluster i, with +/- 50ms tolerance windows. The individual cluster-wise F-measures involved in the resulting average computation are weighted by the proportion of events appearing in that cluster. This enables us to reduce the contribution of unused or scarcely-used timbre clusters.

Information-theoretic viewpoint: expectation entropy

For each incoming event, the entropy of each expectation can provide information about the certainty of the returned prediction. For each predictor, the entropy can be computed as follows:

$$H(p) = -\sum_{1 \leq i \leq K} p(e_i) log_2 p(e_i) \tag{6.15}$$

where $p(e_i)$ is the next event estimated probability distribution over the set of possible symbols K. For each detected event, a number of expectations can be generated depending on the expectation scheme. The variation of the prediction entropy signal may give additional information regarding the structure of the attended stream. We will go back to this aspect when presenting the results in Section 6.7.7.

6.7.2 Experiment: Loop following

Based on the performance metrics introduced above, we present an empirical evaluation based on a database of drum loops. These drum loops provide a good evaluation material because they contain strokes with different acoustic properties, which are organized through time patterns of varying complexity. The task, material and system settings are presented, then we present and discuss the results. The task we propose to simulate is the following: the system is initialized and it does not access any previously trained models; it is exposed to a drum loop repeated several times; once the system has accumulated enough information to perform the bootstrap step and both timbre and IOI symbols are defined, the transcription and expectation timelines are produced. From these, we can compute, at each loop repetition, the statistics presented in previous section.

Material We have selected a subset of audio recordings from the ENST-Drums database (Gillet and Richard, 2006). We use polyphonic drum loops played by one drummer, namely drummer #2 in the database. The database consists of 49 drum excerpts (called phrases in the database) of 9 different styles, for a total of 5627 events. Most of the database patterns have a duration of approximately 10 seconds, that is, slightly more than the time needed to perform the bootstrap computation. In order to observe the learning dynamics of the system, we need to work with longer excerpts. We therefore have edited and looped the original signals 2, 4, 8 and 10 times, depending on the experiment.

Annotations preprocessing As stated by Gillet and Richard (2006), each drum excerpt was annotated following a semi-automatic process. For instance, each cymbal has a distinct label and there is a distinction made between open hi-hat and closed hi-hat. Because each drum excerpt was recorded using 8 microphones, simultaneous strokes were also precisely annotated. The average number of timbre labels present in each excerpt included in our subset is 5.5.

Because our system is processing the input stream with a fixed frame rate (the time precision is 11.6 ms) and performs timbre clustering as if the input was monophonic, we decide to apply the following ground truth preprocessing. If consecutive onsets follow one after the other so that they are in the same analysis frame, we merge the annotated labels into one joint label and keep as onset time the first onset of the consecutive attacks.

Experimental setting The system is designed to learn in a causal way, therefore it may learn more accurately when being exposed to the excerpts several times: this is why we have experimented with a number of repetitions of the basic loop of 2, 4, 6, and 8. The other parameters (e.g. onset detection threshold, model selection information criterion, timbre PCA explained variance, next event predictor context size) are varied for comparison purposes as explained below. Unless explicitly notified, the default parameters we use are listed in Table 6.1.

Parameter	Value
Descriptor set	MFCC
Onset detection threshold	0.6
Onset detection Method	Dual
PCA explained variance	0.6
# Bootstrap events	40
Maximum N-gram order	2
Model Selection Criterion	AIC
On-line K-means η	η^{opt}

Table 6.1: Default parameters used for simulations

6.7.3 Results

This section presents the results of the evaluation whose details are presented in the previous section. Because our expectation system is made of several components, we first present the evaluation of these components in isolation. Then, we provide an overall system evaluation based on the system transcription and expectation accuracies.

6.7.4 Evaluation of system components

It is possible to evaluate some of the system components by providing them as input the data which can be extracted from the ground truth. Given the ground truth annotations (i.e. onset times and labels), we can evaluate some processing stages of the system, such as onset detection and timbre clustering, in isolation. Other stages cannot be directly evaluated with the annotations. The ENST-Drum data is not beat-marked, consequently we have not performed an evaluation of the beat tracking component, and refer to Brossier (2006) for an evaluation of this component. Also, some components rely on a symbolic representation of time. This is the case of the IOI clustering component which provides IOI cluster labels as output, and the next-event prediction component, which uses IOI cluster labels as input and output. Concerning the symbolic expectation component, the PPM stage has been evaluated using symbolic pitch sequences by Pearce and Wiggins (2004). In our case, the symbolic representation we are going to feed to the PPM will possibly be noisy, due to event detection and clustering errors. However, we are interested in evaluating this component under this noisy representation rather than in isolation as shown is Section 6.7.5. We first present the results of the onset detection and timbre clustering stages, when evaluated against ground truth annotations.

Onset Detection

We have first performed an evaluation of the onset detection component by comparing the detected onsets with ground truth annotations, for three onset detection functions labeled *complex*, *dual*, and *hfc*. We refer to Section 6.4.2 for an overview of these methods. For each method, we have varied

the threshold of the peak-picking stage. We used the database presented in Section 6.7.2, and found that the best configuration was associated with the *dual* detection function and a peak-picking threshold value of 0.6. The corresponding F-measure is equal to 0.76, and is comparable to the results obtained by the dual detection function in MIREX 2006 [1]. However, the results are lower than those of more recent onset detection techniques evaluated on drum solos in MIREX 2009 [2]. Nevertheless, here we aim at showing that the possibly noisy representation can still be exploited in further stages of the system. In next section, we study how the automatic detection stage affects the timbre clustering process.

Timbre Clustering

We report here the results obtained when evaluating the timbre clustering component. We propose two evaluation settings in which the inter-onset regions processed by the timbre clustering component are obtained differently. In the first configuration, the regions correspond to the onsets provided by the ground truth of the ENST-drums database. Thus we evaluate the timbre clustering component in isolation. In the second configuration, the audio regions correspond to the onsets computed by the onset detection component, consequently we evaluate the overall accuracy of the combination of onset detection and timbre clustering components.

The timbre clustering accuracy is largely influenced by the outcome of the bootstrap process, in which a PCA can be trained on the bootstrap data, and the estimation of the optimal number of clusters is performed. For this reason, we vary the PCA desired explained variance between 0.1 and 0.8 (or do not perform PCA at all), and the information criterion to be used (either BIC or AIC).

Following, we present the mean transcription statistics obtained when varying these parameters. For each run, we report the Transcription Incremental F-measure (TIFM), the class to cluster ratio (CCR) and percentage of estimates, and the percentage of estimates leading to one timbre cluster (P1). In Table 6.2 the onset times from the ground truth annotations are used, which enables to appreciate the accuracy of system based on a perfect onset detection process. In Table 6.3, the onset times are extracted automatically.

On the one hand, the results corresponding to ground truth onsets show that the combination of the amount of data compression (controlled by the PCA desired variance) and the information criterion play an important role in the outcoming timbre representation. In all cases, the TIFM lies between 0.509 and 0.572. The TIFM average for runs based on AIC (respectively BIC) is 0.551 (respectively 0.536). However we see that AIC-based estimation leads to a better estimation (mean CCR 0.753) than the BIC-based estimation (mean CCR: 0.514). If both criteria tend to estimate less clus-

1. http://www.music-ir.org/mirex/2006/index.php/Audio_Onset_Detection_Results

2. http://www.music-ir.org/mirex/2009/index.php/Audio_Onset_Detection_Results:_Solo_Drum

PCA var./Inf. Cr.	AIC_{GT}	BIC_{GT}
0.1	0.572 0.521 6.382	0.539 0.369 19.178
0.2	0.565 0.519 8.510	0.525 0.353 19.148
0.3	0.570 0.495 4.255	0.539 0.366 21.276
0.4	0.549 0.654 2.127	0.556 0.468 14.891
0.5	**0.559 0.847 0.000**	0.551 0.550 10.631
0.6	0.522 0.896 0.000	0.548 0.635 8.512
0.7	0.536 0.958 0.000	0.545 0.649 8.512
0.8	0.545 0.956 0.000	0.515 0.620 19.143
No	0.545 0.933 0.000	0.509 0.624 19.571

Table 6.2: Timbre clustering statistics for different bootstrap settings depending on the PCA desired explained variance ("No" means no PCA is applied during bootstrap). The columns show the information criterion used. AIC_{GT} and BIC_{GT} corresponds to runs that use the ground truth onsets. For each run, the measures presented are, from left to right, Transcription Incremental F-measure, Class to Cluster Ratio, and Percentage of estimates leading to one timbre cluster.

ters than the ones available in the ground truth, the AIC estimation tends to produce a higher number of timbre clusters. This is reflected in the P1 measure: AIC-based runs estimate a single cluster less frequently than BIC-based runs. Overall, the AIC criterion can be seen as more sensitive than the BIC. From this, the choice of the PCA explained variance is a tradeoff between higher TIFM (low PCA explained variance) or higher CCR (high PCA explained variance). Situations resulting in one single cluster render the clustering process useless, since there is no information gain. The P1 statistic shows that the BIC-based estimation leads to a unique timbre cluster in 20.21% of the runs in average, while the P1 of AIC-based runs has a mean of 0.6%. Additionally, by comparing Tables 6.2 and 6.2, we can see that when ground truth onsets are used, the resulting TIFM values are slightly higher, the other parameters being fixed. This shows that when the errors made by the detection stage are indeed propagated into further stages of the system. However, the TIFM statistics follow the same trend in both settings.

For the subsequent experiments, we choose to perform the bootstrap estimation with the AIC criterion and a PCA explained variance of 0.6, as a tradeoff to maximize both TIFM and CCR.

6.7.5 Expectation

After the system has completed the bootstrap step it generates expectations. From this moment, we can evaluate how generated expectations match both transcription and ground truth. We now present the distribution of the expectation statistics obtained when attending each excerpt of the ENST subset as described in previous section.

PCA var./Inf. Cr.	AIC_{TR}	BIC_{TR}
0.1	0.572 0.521 2.127	0.539 0.369 29.787
0.2	0.565 0.519 2.127	0.525 0.353 34.042
0.3	0.570 0.495 2.127	0.539 0.366 31.914
0.4	0.542 0.659 0.000	0.539 0.441 21.276
0.5	0.519 0.819 0.000	0.550 0.579 19.148
0.6	**0.526 0.904 0.000**	0.557 0.590 17.021
0.7	0.516 0.899 0.000	0.553 0.569 23.404
0.8	0.522 0.883 0.000	0.558 0.475 25.531
No	0.502 0.891 0.000	0.556 0.469 26.341

Table 6.3: Timbre clustering statistics for different bootstrap settings depending on the PCA desired explained variance ("No" means no PCA is applied during bootstrap). The columns show the information criterion used. AIC_{TR} and BIC_{TR} correspond to runs using the detected onsets. For each run, the measures presented are, from left to right, Transcription Incremental F-measure, Class to Cluster Ratio, and Percentage of estimates leading to one timbre cluster. It can be seen that the overall TIFM values are slightly lower than those obtained when ground truth onsets are available.

Influence of exposure

In this experiment, we aim to show the impact of exposure in the system predictive accuracy. Our initial guess is that the system is sensitive to exposure, but we aim at quantifying how each expectation scheme is sensitive to repeated patterns. For each expectation scheme, we report in Figure 6.8 four independent runs in which we present to the system each drum pattern repeated 2, 4, 6 and 8 times. All schemes are characterized by a high EIP (greater than 0.65 in all cases) and a lower EIR (lower than 0.42 in all cases).

As expected, when the number of repetitions increases we observe an increase of all expectation statistics. In the case of the joint and independent schemes, the WFM has the biggest increase, which means the PPM expectator can take advantage of the repetitions to learn to provide a prediction which matches its transcription. The other expectation statistics, which are related to the ground truth, also increase -to a lesser extent- with an increasing number of repetitions. The when|what scheme expectation statistics are less affected by an increase in the number of repetitions. For this scheme, the EIFM decreases when the number of repetitions goes from 6 to 8. Overall, the results show that both joint and independent schemes enable the system to create a internal representation which sequential regularities can be learned, as shown by the WFM statistics. The WFM increase is less marked when using the when|what scheme, suggesting that the resulting internal representation does not contain such regularities. A reason of this may be that the when|what scheme representation is not as robust to transcription noise. We will go back to this in Section 6.8.

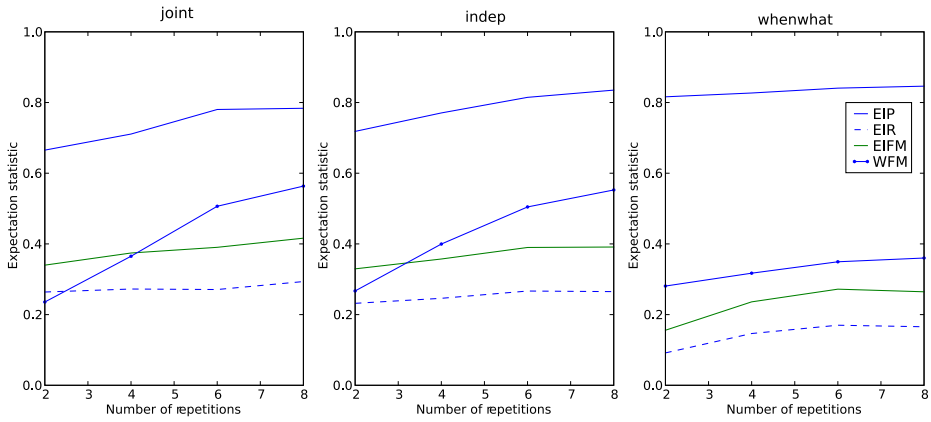


Figure 6.8: Comparison of expectation statistics (EIP, EIR, EIFM, WFM) as a function of the number of repetitions of a given loop. The three expectation schemes are compared. Left: joint scheme, middle: independent scheme, right: when|what scheme.

Influence of context size

As the PPM predictor we use to provide a prediction of the next event is based on the observation of a context of fixed size N (see Section 6.6.1), we are interested in measuring the impact of the prediction context size. Indeed, the number of past items involved in the posterior probability computation may affect the behavior of the learner by biasing it with predictions which are too general (low N), or by overfitting the prediction (high N). In Figure 6.9 we show the average expectation statistics (EIP, EIR, EIFM and WFM) plotted against the size of the context length N, which is varied from 1 to 6.

Overall, the joint scheme leads to the highest EIR and EIFM while the independent scheme leads to the highest EIP. Both independent and joint schemes exhibit a similar dependency to context size. Both models exhibit an increase of the WFM when the context size varies from 2 to 4, and the independent scheme WFM grows along the context size. This is not the case of the when|what scheme, in which the expectation statistics exhibit almost no variation with an increasing context size. Two factors may explain this: first, the when|what scheme intrinsically needs less context to encode patterns of rhythmic events, because the timing structures between events *of the same timbre category* may be simpler to describe than those between successive events regardless their timbre category. Consequently, structures associated to the when|what scheme need less context to be learned. Moreover, whereas this first reason holds for an error-free representation of timing structures, here the unsupervised transcription stage provides a noisy representation of events. Events labeled with the wrong timbre category will give rise to erroneous cluster-wise inter-onset intervals, which form the basis of

the when|what scheme encoding. Overall, the when|what prediction scheme may appear as a processing path that is less robust to transcription error than the independent and join schemes.

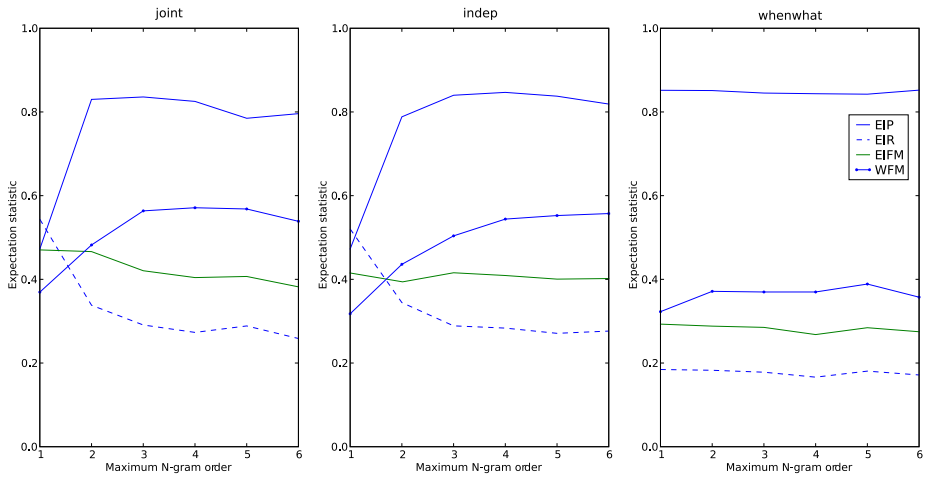


Figure 6.9: Comparison of expectation statistics (EIP, EIR, EIFM, WFM) as a function of the maximum order used to provide a prediction of the next event. The three expectation schemes are compared. Left: joint scheme, middle: independent scheme, right: when|what scheme.

6.7.6 Expected onset detection

Because our system representation is unsupervised and the number of timbre clusters is not fixed, it is not possible to directly compare the performance of the transcription and expectation stages with existing systems. However, we can can collapse the timeline of expected events into a train of onsets, and compare these expected onsets with the ground truth using standard evaluation techniques.

We report in Table 6.4 the results of comparing the train of expected onsets, with the ground truth onsets. The parameters we use are the default values presented in Table 6.1. These figures can be compared with the results obtained in the onset detection evaluation in Section 6.7.4. The joint and independent schemes lead to an expected F-measure which is above 0.6, that is, about 15% lower than the onset transcription F-measure. The when|what scheme leads to poorer onset expectation results.

Overall, the evaluation reported here shows that the expectation schemes have distinct behavior with respect to the variations of the expectation statistics depending of exposure and context size. While the joint and independent schemes have a similar behavior (the joint scheme slightly outperforms

joint	indep	when|what
0.613	**0.619**	0.424

Table 6.4: F-measure, computed between expected events and ground truth annotations, as a function of the expectation scheme.

the independent scheme in most of the cases) and dependency to exposure and context size, the when|what scheme exhibits worse performance and a smaller dependency on exposure and context size. These findings will be addressed in the discussion.

6.7.7 Expectation entropy and structure finding

We would like to analyze the time dynamics of the expectation signal. Our aim is to assess if the expectation signal can provide an account of the attended excerpt structure. To do this, we compute the instantaneous prediction entropy during the analysis.

In Figure 6.10, we compute the entropy (following Equation 6.15) of both BRIOI and timbre predictors when processing the commercial drum'n bass excerpt we used as a running example. The basic loop boundaries, which are unknown by the system, are shown using red vertical lines. We observe an overall decreasing trend in the entropy curve. The basic loop consists itself of four variations of the same rhythmic pattern, and this internal structure appears plotted in the figure.

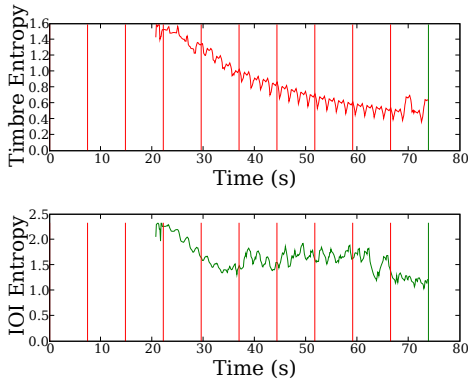


Figure 6.10: Instantaneous Entropy of timbre (top) and BRIOI (bottom) predictors for a commercial drum'n bass excerpt. The entropy signal appears after 20s, when the bootstrap step has been performed. Globally, the entropy displays a decreasing trend that corresponds to the learning of the excerpt. Locally, we can see repeating patterns reflecting the loop structure.

It is worth noting that the prediction entropy can be useful for describing other signals than drums loops. In the following example we use the joint

expectation scheme to process a polyphonic piano recording, made of two repeated parts, which are themselves made of two variations of the same motive. We show in Figure 6.11 the instantaneous prediction entropy when attending the piano excerpt repeated 8 times. After a the third repetition, the entropy pattern adopts a shape that reflects the structure of the excerpt, with high entropy values at the beginning of motives and low entropy values marking the beginning of the sub-motives. This pattern may be compared with the error pattern that is obtained by Elman (1990) when predicting sequences of words (see Figure 3.7). After the sixth repetition, the entropy pattern is altered, local entropy minima indicating motif boundaries become local maxima. However, the entropy pattern keeps the periodicity that reflects the two motive repetitions.

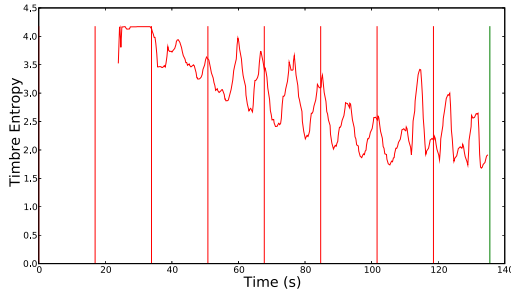


Figure 6.11: Instantaneous Entropy of the combined Timbre-IOI predictor for a piano recording. The entropy signal appears after 20s, when the bootstrap step has been performed. Vertical bars indicate repetitions of the excerpt, unknown by the system. After a few repetitions, the entropy pattern settles and adopts a shape that reflects the structure of the excerpt.

Overall, we have shown here that the information of prediction entropy dynamics may be used to reveal the overall structure and redundancies is the analyzed signal. This finding may lead to enable the system to perform *on-line structural segmentation* tasks. However, the scale of the structures to be highlighted would need to be controlled to tune the system to specific tasks. Moreover, as we have seen in Figure 6.11, the entropy signal pattern seems to stabilize through several repetitions of an excerpt and is not stable after a certain number of repetitions. However the results shown here are promising as they show that a causal prediction-oriented listening system can implicitly perform analysis tasks that are usually done by ad-hoc, offline systems (Foote, 1999; Aucouturier and Sandler, 2001; Peeters et al., 2002; Goto, 2003; Ong, 2006).

6.8 Discussion

We have presented a system that addresses simultaneously the detection of temporal and timbre events, their categorization, and the prediction of

forthcoming ones. The idea of describing the what/when musical stream through the formation of a set of time and timbre categories is rooted in experimental findings, as presented in Chapter 2. Here, we have discussed 3 different representation and expectation architectures, namely independent, joint, and when|what schemes. Some of them having more physiological and cognitive plausibility than others. Independently of the used architecture, we have defined two main processing constraints, namely unsupervised learning and causal processing. In our approach, we do not cope with an incremental learning approach which is capable of constantly modifying itself by adding or removing categories of timbre or durations. Instead, during a bootstrap phase of the system, the number of clusters is estimated and they are initialized, before generating expectations. However, the constraints make our approach distinct to supervised approaches to timbre transcription, where the target timbre categories are known in advance. Our approach can also accommodate a supervised transcription setting, as we will show in Section 6.8.

For validating the system in a unsupervised way, we have defined new measures to evaluate both unsupervised transcription and unsupervised expectation tasks. We have run an evaluation of two system components (onset detection and timbre clustering) and the combination of those two, which represent the unsupervised transcription. These experiments have shown that the way the information is represented (e.g. by varying the PCA explained variance) and the information criterion we use influence the timbre events representation. The Incremental F-measure proposed by Marxer et al. (2008) is a means of evaluating how close the set of symbols is to the ground truth labels. However, additional measures are needed to assess the sensitivity of the system. The Akaike information criterion does not penalize the complexity as much as the Bayesian information criterion does. We find that the Akaike criterion coupled with a moderate compression of the timbre features (i.e. PCA explained variance of 0.6) leads to the best tradeoff between complexity and similarity with the ground truth.

In our approach, the onsets phase is not explicitly encoded: the time location of events relative to the previous beat is not taken into account, which makes beat-tracking an optional process. Rather, inter-onset intervals is the only timing information that is used. Even in this case, as noted in Section 6.7.7, the expectation entropy signal may create an implicit phase signal corresponding to the location of current event in a higher-level temporal structure. Thus, our IOI-based time representation may be seen as a complementary way of generating expectations compared to approaches that explicitly rely on extracted beat, beat weight, or metrical hierarchy (Smith, 1996).

Furthermore, in none of the evaluated configurations we have been able to reach an Expectation Incremental F-measure higher than 0.572 for the evaluation dataset. For reference, the best performing MIREX 2005 drum transcription entry (Yoshii et al., 2005), was evaluated to have an average F-measure of 0.659 for a three-class transcription task. But in this latter work, the task was to generate a transcription using a predefined set of classes. Because our system performs unsupervised transcription and expectation, the

evaluation measures differ and these figures cannot be directly compared. However, it would be a natural extension to our work to characterize the performance of the expectation system in the context of supervised transcription. This would allow us to directly compare the expectation statistics with the transcription results of existing systems. To increase the transcription accuracy, we aim at investigating how to combine a short-term representation of timbre and time (e.g. bootstrap estimation) with a longer-term representation which may involve a database of predefined timbre categories.

Concerning the performance of the whole expectation system, we can make a distinction between the independent and joint schemes, for which expectation performs in the same order of magnitude and depends on repetitions and context size, and the when|what scheme, for which expectation performs worse and depends less on training duration and context size. To explain this, we assume that the accuracy of the when|what scheme is more crucially altered by errors in the transcription, because these errors generate in turn errors in the representation of timing events by creating erroneous clusters of inter-onset intervals. From a computational point of view, these architectures may be seen as parallel processing paths. The independent and joint schemes need more information to be stored (because transitions between timbre events are also encoded). The joint scheme space requirements are higher because the transitions between all combinations of timbre and time symbols are stored. From a musical point of view, the independent and joint schemes would be able to code information about the musical surface, such as melodies or drum solos.

Contrastingly, the transitions between timbre events are implicitly coded in the when|what scheme. This makes its space requirements low when representing rhythms (e.g. the time dependencies between onsets with same timbre are simple, even if the sum of onsets over timbre forms a more complex structure). However, to be properly applied to musical audio signals, this scheme requires the transcription component to perform efficiently. The transcription accuracy may be increased by using source separation (e.g. via independent component analysis or non-negative matrix/tensor factorization) instead of merging together simultaneous attacks. Also, the use of a polyphonic detection model, in which two distinct timbre categories can be detected at the same time, may make the when|what scheme more competitive.

Whereas the independent scheme of inter-onsets captures rhythmical aspects encoded as durations between onsets, the independent scheme of timbres encodes the regularities in the pure order of the events abstracting from specific durations. The plausibility of the when|what scheme versus the independent schemes depends on the degree of streaming. If a strong tendency towards streaming yields the perception of separate synchronous rhythms (for each particular percussion sounds, e.g. the hi-hat or bass drum rhythm in isolation), the when/what scheme is preferred over the independent scheme. The more interdependent the sound classes and their durations (more precisely: IOIs) the more appropriate the joint scheme. Overall our system could take advantage of combining these three schemes, which represent different statistical and musical viewpoints for pattern matching and

expectation tasks.

Using supervised detection models In this chapter, we have emphasized the use of a unsupervised representation component, to show how a general-purpose model can process distinct types of musical audio excerpts. However, it is also possible to use a supervised transcription component, and to plug-in this component into the expectation model. For instance, to handle a polyphonic representation, we can use a set of binary classifiers (one for each sound to be represented) in the representation layer. Therefore, we can adapt the representation architecture corresponding to the when|what prediction model depicted in Figure 6.6. The resulting architecture is shown in Figure 6.12.

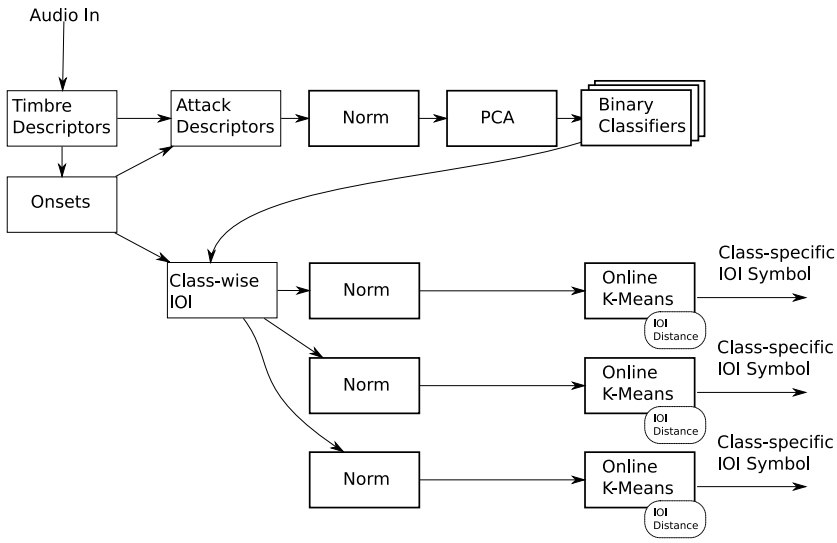


Figure 6.12: Block diagram of the representation layer used for the when|what in a supervised setting. We refer to the legend in Figure 6.5

The work presented here emphasizes the expectation as being a central process involved in music listening. The ability of a system to form expectation may serve as a measure of musical complexity. Expectation describes structure by inducing a segmentation through points of high or low expectation. Finally, in the context of causal modelling, we can see the expectation as a dynamic top-down control which may modulate lower-level processes such as onset detection. In our what/when prediction framework, the expectation feedback may be provided to timbre-specific event detectors. In the context of musical audio analysis, we view models of music expectation as general components able to dynamically accumulate the structure of the

drum loop	# perc.	# comb. perc.
simple disco	5	5
complex funk	5	12

Table 6.5: Number of percussion sound classes (denoted perc.) and number of combinations of simultaneous classes (denoted comb. perc.) in the two examined drum patterns.

musical environment, and where the expectation signal may help to solve more specific musical tasks.

6.9 Examples

Let us now illustrate the working of our system with two examples from the ENST data base. The two drum patterns have contrasting degrees of complexity. In the ENST data base they are called *phrase_ disco_ simple_ slow_ sticks* (*simple disco*) and and *phrase_ funk_ complex_ fast_ sticks* (*complex funk*). No source separation is performed as a preprocessing step. Therefore, it should be considered how many different percussion sounds appear in the drum samples and how many different combinations appear (Table 6.5), since each sound combination may lead to a different cluster. In our examples, we observe that estimated cluster numbers are less than the numbers of percussion sounds. We have calculated the matching matrix between the annotated onset events ('score') of a class (e.g. chh_bd=closed hi-hat and bass drum played synchronously) and the detected onsets of a cluster that has emerged in our system. In this matching matrix we can iteratively yield the maximal entry thereby establishing a connection between a row (class) and a column (cluster). After elimination the row and column of the maximal entry we determine the maximal entry again until the matrix vanishes. This procedure endows us with an optimal mapping between the classes and the clusters. In Figures 6.14 and 6.16, we display sequences of classes and clusters on the same line if they are interconnected through this mapping. For the *simple disco* pattern (Figure 6.14), it can be seen that fragments of the basic pattern bass/open hi-hat/snare/closed hi-hat are captured. The single cymbal instance is not captured. However, one cluster (second highest row) can be interpreted as detecting the sustain phase of the cymbal (three hits).

Considering the *complex funk* excerpt, we show in Figure 6.15 the first half of the looped pattern. The score expressed as combination of instruments, the sequence of detected timbre clusters and the expectations are shown in Figure 6.16. The number of extracted clusters (six) is less than the order of occurring combinations of percussion sounds (twelve). Several sound combinations occur sparsely. The mapping between sounds and clusters is not clear. The expectations cannot capture the complexity of the pattern well.

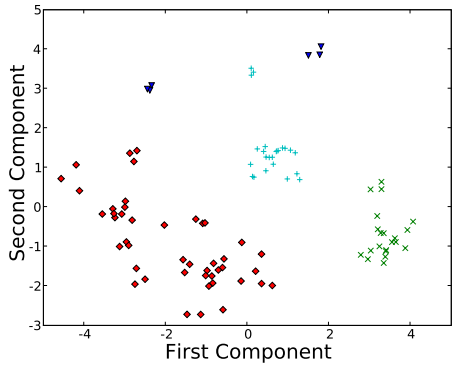


Figure 6.13: Acoustic properties of the detected attacks, plotted along their 2 first principal components. Colors and shapes indicate timbre clusters assignments.

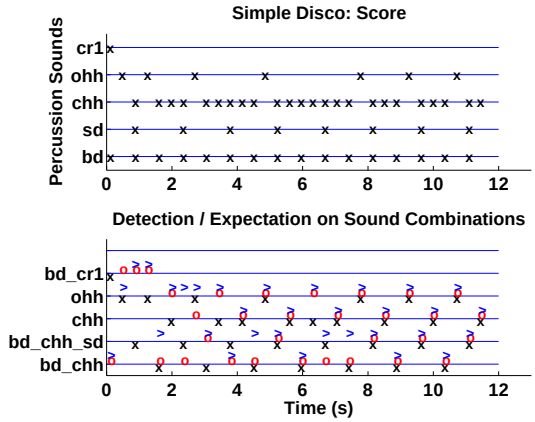


Figure 6.14: Top: score of the *simple disco* excerpt. Bottom: Score represented as combinations of instruments ('x') , extracted cluster sequence ('o') and expected events ('>') according to the independent scheme (bottom) for this excerpt. This acoustic pattern is learned well after processing the first 8 s of signal.

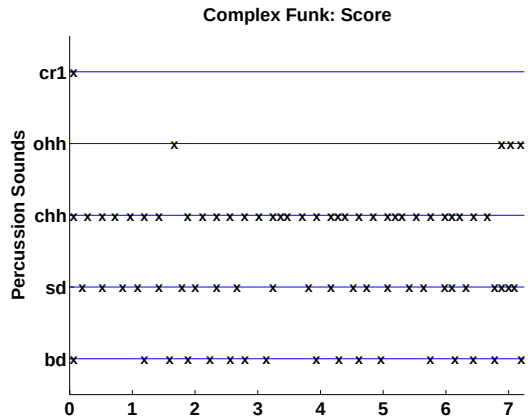


Figure 6.15: Score of the *complex funk* excerpt.

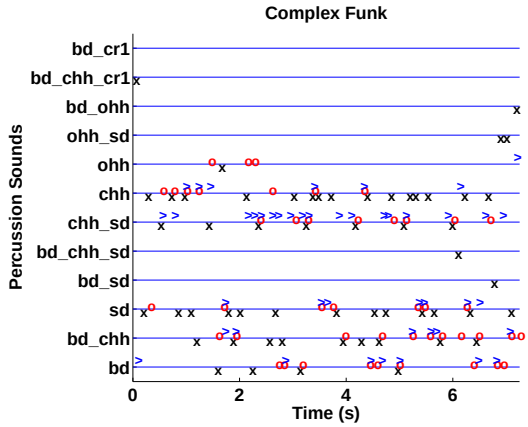


Figure 6.16: Score represented as combinations of instruments ('x') , extracted cluster sequence ('o') and expected events ('>') according to the independent scheme (bottom) for the *complex funk* excerpt. The system fails in representing adequately this irregular excerpt and cannot generate useful expectations.

CHAPTER 7

Integration in Music Processing Systems

7.1 Chapter Summary

The model presented with this thesis has been integrated in different musical systems, each one highlighting specific aspects of it. The representation layer has been integrated into a real-time music transcription system. The combination of representation and expectation modules has been the basis of a software tool able to visualize and evaluate the ongoing representation expectation of an attended signal, which was developed within the Emcap project [1]. Also, this tool allows the sonification of the expectation process, that is, the generation of an audio rendition of the expected events by means of concatenative synthesis, in a fully unsupervised setting. Sonification provides a way of inspecting the internal dynamics of the what/when expectation model. Finally, a real-time application has been designed to provide a real-time visualization of the system expectations.

7.2 Integration of the Representation Layer

7.2.1 A library for high-level audio description

Various aspects of the representation layer described in this thesis have been implemented in a C++ library called Billaboop [2]. More generally, Billaboop is a library which provides high-level interaction with audio, and was designed to be included in real-time audio applications. On the one hand, Billaboop is similar to the Aubio [3] library because it is intended to be a low-latency library for causal and possibly real-time processing. On the other

1. http://emcap.iua.upf.es/
2. http://billaboop.com/lib
3. http://aubio.org

hand, Billaboop takes advantage of of a more structured and object-oriented design in C++, which makes it similar to the libraries CLAM [4], Marsyas [5], or Essentia [6]. Moreover, Billaboop aims at integrating audio analysis, data modelling and machine learning tools in a single library.

First, the library provides a common framework for computing audio descriptors either in the temporal or the spectral domain. Each low level descriptor processing class is derived from the abstract Descriptor class. This makes it easy to add new descriptors to the applications, and to add new descriptors to the library analysis processing queue. The low-level descriptors available so far are listed below:

- Zero Crossing rate
- Complex Domain Detection
- High Frequency Content
- Spectral Centroid
- Spectral Slope Regression
- Energies in Bark, Gammatone, and Mel filterbanks
- Mel-frequency Cepstrum Coefficients

The library also provides a set of mathematical and statistical tools that enable to derive new descriptors from the base set. These tools include methods such as linear regression, data normalization or peak picking. This enables to easily build higher-level descriptors such as temporal regression over the instantaneous energy of a given filterbank or onset detectors. Higher level descriptors, resulting for a categorization process, can be extracted with the following machine learning algorithms.

- K-Nearest Neighbours
- Support Vector Machines (SVM, using a wrapper to the libsvm [7] library)
- K-means clustering
- Online K-Means

The supervised algorithms can be run in testing mode based on previously built models. However, it is also possible to train models from the library, which is useful from running validation experiments or to provide embedded learning in applications that use the library.

The Billaboop library thus provides supervised or unsupervised detection of acoustic events based on arbitrary descriptors. The resulting higher-level description of the incoming audio stream enables to perform the following tasks:

- Query by audio similarity
- Query by cluster prototype
- Concatenative synthesis

Billaboop is available under the GNU general Public License (GNU GPL). It has been tested on Windows, MacOSX and Linux environments. In the next section we present a front-end to the transcription layer provided by this library.

4. http://clam-project.org
5. http://marsyas.sness.net
6. http://mtg.upf.edu/technologies/essentia
7. http://www.csie.ntu.edu.tw/~cjlin/libsvm

7.2.2 Real-time interaction front-end

A real-time application called *Billaboop Drums* has been implemented to provide a virtual drumming interface. The main idea is to provide to users a real-time transcription of percussive strokes detected in the signal input. The user can choose to use a predefined transcription model or to build one within the application. This makes it possible to use arbitrary sounds (tool drumming, hand clapping, beat boxing) to control the application. As a result, the application returns either a MIDI transcription of the performed rhythm or can control an internal sampler by performing query by similarity.

A screenshot of the Billaboop Drums VST plug-in is shown in Figure 7.1. The vertical slider controls the onset detection sensitivity. Three target classes are defined, namely Bass Drum, Snare Drum and Hi Hat. The radio buttons *auto*, *wizard*, and *edit* enable to choose the sound detection recognition mode. When running the *auto* recognition mode the programs uses a previously built, general purpose sound classification model. As reported in (Hazan, 2005b), this model has been trained on a dataset using sources from professional and amateur beatboxers, as well as arbitrary sounds (e.g. keys, spoon hitting a tea mug, handclaps). The accuracy of this model has been discussed in (Hazan, 2005b), and showed that a cross-validation accuracy of 81% can be attained if the system has to return a decision in 11ms. This makes this general model usable for real-time performance especially because higher detection rates can be attained when a given user gets used to the predefined model. However, in other cases it is preferable to allow the user to build its own recognition model. This can be achieved on-line, using the *edit* or *wizard* mode. In this case, the user provides a number of exemplar acoustic strokes for each drum label. Then, the embedded learning component provided by the Billaboop library enables to train a recognition model. The wizard mode makes it faster to provide exemplar strokes by guiding the user with spoken instructions without the need of using the graphical user interface.

7.3 Analyzing, evaluating, and sonifiying representation and expectation

We present in this section the command line tool used to perform expectation generation experiments using the what/when expectation model presented in this thesis. This tool is called Billabio, and was first introduced in (Hazan et al., 2007). Billabio is implemented in the Python [8] programming language using the Numpy package [9]. The resource-intensive operations such as audio analysis are handled by two external libraries, namely Aubio and Billaboop. The general system block diagram has been presented in Figure 6.1. Additionally, detailed diagrams specific to a given expectation scheme have been shown in Figure 6.5 and 6.6.

8. `http://python.org`
9. `http://numpy.scipy.org`

Figure 7.1: Screenshot of the Billaboop Drums VST plug-in.

The general idea behind this tool is to study the dynamics of the what/when expectation system by providing visualization, evaluation and sonification of the expectation process. The audio analysis stage corresponding to feature extraction can be parametrized, allowing to choose from a set of descriptors. The visualization module has been used to produce all the figures presented in previous chapter. The evaluation can be performed either in an unsupervised or supervised settings. In the latter case, annotated files corresponding to each of the analyzed audio files need to be provided. The behaviour of the system modules can be configured. We show in Table 7.1 the list of command-line parameters that can be passed to the application. We report the main parameters that can be controlled. First, the user launches the analysis process by providing an audio file playlist. Indeed, because we use a dynamic listening model, the ordering of the audio sources is important. The number of repetitions can be provided. Then, the auditory front end can be configured by choosing the onset detection thresholds and technique used, the list of descriptors to use and the descriptors averaging technique, if any. The dimensionality reduction module can be controlled by tuning the PCA process and the online K-means clustering parameters. Finally, the expectation module can be configured: the maximum context size to considered by the PPM expectators and the expectation scheme can be chosen.

```
Usage: ./billabio [options] soundfile [soundfile_2 soundfile_N]

Options:
  -h, --help            show this help message and exit
  -B BUFSIZE, --bufsize=BUFSIZE
                        buffer size [default=1024]
  -H HOPSIZE, --hopsize=HOPSIZE
                        overlap size [default=512]
  -g NGRAMSIZE, --cngramsize=NGRAMSIZE
                        the size of the n-grams [default=5]
  -s EXPECTATORSCHEME, --expscheme=EXPECTATORSCHEME
                        the expecator scheme [default=indep]
  -t DETECTIONTHRESHOLD, --threshold=DETECTIONTHRESHOLD
                        onset detection threshold [default=0.5]
  -p DOPLOT, --plot=DOPLOT
                        whether to plot the results on a separate window
                        (file/window/no) [default=no]
  -S, --sonify          whether to sonify the transcription and the
                        expectation
  -r NREPEATS, --repeat=NREPEATS
                        how many times to repeat the input sound [default=1]
  -d DESCMODE, --descmode=DESCMODE
                        whether to store each descriptors on the onset frame
                        or to compute the median the IOI region
                        (onset/ioi/onset2) [default=ioi]
  -D DESCLIST, --desclist=DESCLIST
                        which timbre descriptors we aim to use. The format is
                        "desc1 desc2", where the descriptors have to be chosen
                        from the keywords: sc, zcr, pitch, mfcc  [default "zcr
                        sc pitch"]
  -i INFOCRITERION, --infocriterion=INFOCRITERION
                        Model selection criterion (bic/aicc) [default aicc]
  -e IOILRATE, --ioilrate=IOILRATE
                        ioi online k-means learning rate [default .3]
  -f TIMBRELRATE, --timbrelrate=TIMBRELRATE
                        timbre online k-means learning rate [default .3]
  -X, --pca             whether to do PCA on timbre descriptors [default no]
  -E, --eval            whether to evaluate attended excerpt against ground
                        truth, for each loop
  -P DBPATH, --dbpath=DBPATH
                        Database path, used for evaluation
```

Table 7.1: List of configurable parameters in the Billabio application

7.3.1 Implementation

The components of Billabio are written in the Python programming language, which the exception of the audio analysis processes, which are implemented in C. Both dimensionality reduction and next event prediction modules, constituting the machine learning algorithms, are implemented using numpy. Additionally, we use the em package [10] (Cournapeau, 2006) for

10. http://www.ar.media.kyoto-u.ac.jp/members/david/softwares/em

applying the Expectation Maximization Algorithm, and the mdp package [11] (Berkes and Zito, 2007) for performing the Principal Component Analysis.

7.3.2 Sonification

Additionally, the system enables to sonify the transcription and expectation processes, as explained below. Expectation sonification provides a simple but powerful tool for evaluating qualitatively the system dynamics for encoding an audio input and generating an expectation signal. Basically, we design a system, inspired by Schwarz (2004), Collins (2004) and Jehan (2005) based on data-driven concatenative synthesis, as explained below. During the listening process, each inter-onset region is described using some features (e.g. Zero Crossing Rate, Centroid, MFCC, Pitch). The inter-onset region description is associated to a timbre cluster and the waveform corresponding to this region is stored.

Consequently, we can resynthesize the transcription based on the accumulated slices and events cluster assignments. The expectation module provides a timeline corresponding to the future onset times for each timbre cluster. Based on this timeline, we retrieve, for each expected cluster, the waveform which is prototypical for this cluster. This way, we can generate an output audio stream which sonifies both encoding and expectation processes. We include examples of expectation sonification for various audio files, either containing pitched or percussive sounds, in Appendix B.

7.4 Real-Time What/When Expectation System

The real-time *what/when* expectation system(Hazan et al., 2008a) is an application demonstrating the real-time use of the expectation model developed in Chapter 6 which illustrates the when|what prediction scheme. As noted in Section 6.8, this prediction scheme enables the processing of a polyphonic representation of events. Because this scheme is more sensitive to the transcription noise than the joint and independent schemes, it requires an accurate polyphonic detection model. The online K-means algorithm we introduced in previous chapter cannot model the occurrence of simultaneous strokes for different categories, for this reason we provide here a supervised detection model that can handle a polyphonic representation of percussive strokes. The block diagram corresponding to a supervised polyphonic detection model is shown in Figure 6.12. The representation of acoustic units relies on a set of supervised classifiers that were trained on the MAMI database (Tanghe et al., 2005b), which consists of annotated drum samples in a polyphonic context. Our implementation allows the use of arbitrary binary libsvm models for each sound detector. Here we have trained four binary classifiers to detect the occurrence of the following labels: Bass Drum, Snare Drum, Open Hat, and Closed Hat. The binary classifiers are SVM models trained using the libsvm wrapper that is included in the Billaboop library.

11. `http://mdp-toolkit.sourceforge.net`

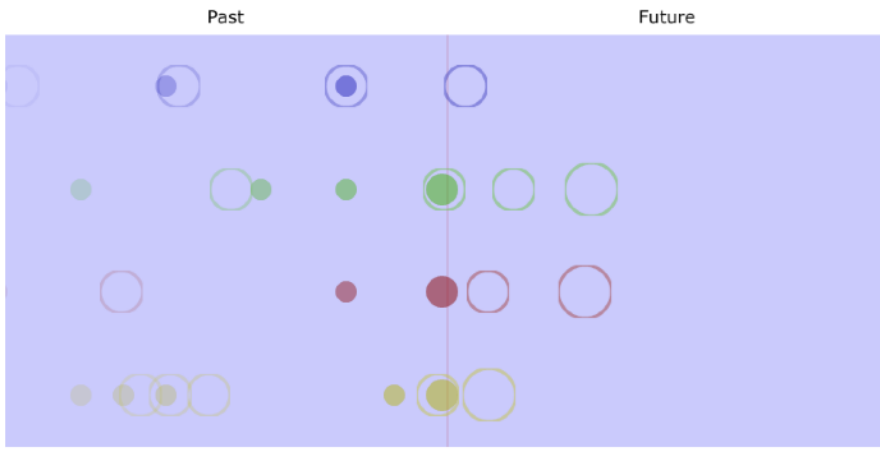


Figure 7.2: Screenshot of the real-time what/when expectation system, analyzing the song *Highway to Hell* (AC/DC, 1979). Four stroke categories are used, and correspond to four rows and colors.

Salient events in the incoming music stream are detected and categorized. For each timbre category, expectation of the future events to be heard is computed. The analysis, sonification, and the visualization of transcription and expectation are done real-time, which enables an interactive user experience. The visualization of musical events is divided into two parts: past (left) and future (right). The vertical red line indicates the present. At each instant, when a perceptual hit is detected, a point appears at the present. Detected events of each category trigger a specific drum sample, in a way similar to the *Billaboop Drums* implementation. The color and row correspond to a specific timbre category. Detected stokes appear as solid colored balls, in the present line and they scroll to the left as the acoustic stream is analyzed. Based on the detected events, expectations are generated. Empty circles corresponding to those expectations are consequently displayed as future events, at the right of the present line. When a detected event matches and expected event, this means that the expectation is fulfilled.

We show a screenshot of the system while processing a commercial recording in Figure 7.2. The recording used for this example is *Highway to Hell* (AC/DC, 1979). The detected and expected events correspond from to to bottom, to the Bass Drum, Snare Drum, Open Hat and Closed Hat detection models. We can see the expectation pattern displayed with empty circles. Some expectations, namely in the Bass Drum, Snare Drum and Closed Hat categories are matched. Also, we can notice in the top left corner a an unmatched expectation for the Bass Drum, that corresponds to an event that was missed by the detection model but was expected by the system, because it was following the regularities of the Bass Drum pattern analyzed so far.

7.5 Concluding remarks

In this dissertation we have stressed the interplay that takes place between representation, learning, and expectation. We have introduced in this chapter different implementations that enable the comparison of design alternatives, and proposed a musical system that allows the use of representation and expectation models for processing spectro-temporal events in musical audio signals. Of course, the representation we use does not allow us to process some important events in music such as chords or harmony, and by no means we claim to have proposed a system that can process all kind of musical signals. As such, these tools and applications have to be seen as a starting point for making representation and expectation models available for processing musical audio, and are aimed to evolve into novel application uses for research, compositional and interaction purposes. In the next chapter, we summarize and discuss the whole approach presented in this dissertation, and present general conclusions.

CHAPTER 8

Conclusions and future work directions

In this dissertation, we have shown that it is possible to build a music representation and expectation model that exhibits basic characteristics found in auditory perception and cognitive systems:

- Causal processing: new acoustic information is processed as it becomes available.
- Adaptation: Incoming information enables the refinement of the internal representation of musical sequences.
- Prediction: The sensory information and current representation of the acoustic stream enables the formation of expectations regarding forthcoming events.

These principles have formed the basic thread from which we have presented and discussed the following contributions.

8.1 Contributions

A prediction-based account of implicit learning We have proposed in Chapter 4 an approach to modelling the acquisition of statistical regularities in tone sequences. We have used two Artificial Neural Network architectures to simulate the general learning trend observed by Saffran et al. (1999). Our results show that the choice of the network architecture has little effect on the post-exposure accuracy, which suggests that an extended temporal context is not necessary to model this task.

The simulations based on interval representations such as Pitch Class Interval or Contour did not consistently account for the experimental results. However, using a tone sequence encoding based on Pitch Class, we observed, for all experiments and languages, an increase of the categorization accuracy of words versus non-words and words versus part-words in a population of

prediction models after they have been exposed to tone sequences containing statistical regularities.

In this learning modelling experiment, our main contribution is two-fold. First, we showed that the choice of the representation of the musical events must be taken into account when modelling music perception in humans and machines, even when working in the symbolic domain. Furthermore, our results suggest that the common assumption that pitch sequences are encoded as abstract interval values cannot be taken for granted. More importantly, we have proposed a framework for simulating and evaluating the outcome of sequence learning experiments based on the forced-choice task paradigm in a way that allows for the comparison of arbitrary learning algorithms. This framework is extensible and enables the assessment of whether learning architectures can simulate further empirical findings using real-world musical sequences (Schellenberg, 1996; Dalla Bella et al., 2003).

What/When expectation model We have presented an unsupervised and causal approach to transcribe, encode and generate *what* and *when* expectations based on constant-tempo musical audio, using both timbre and timing dimensions. Our system analyzes and represents events based on beats and onset, and creates a set of categories, in either an unsupervised or supervised way to describe incoming events. Then, the model learns the regularities of the incoming signal to create expectations. As such, this can be considered as a mid-level approach to expectation modelling applied to audio, if we compare it to systems that focus on auditory sequences (Brown, 1993; Cooke, 1993; Ellis, 1996), information-theoretic approaches that do not explicitly extract semantic information from the musical audio (Abdallah, 2002; Dubnov, 2006; Cont, 2008), and models that focus on symbolic or MIDI representations of music (Bharucha and Todd, 1989; Todd and Loy, 1991; Mozer, 1994; Tillmann et al., 2000; Lartillot et al., 2001; Pachet, 2003; Eck and Schmidhuber, 2002; Pearce and Wiggins, 2004).

Alternatives to combining the *what* and *when* dimensions have been presented and evaluated. We have illustrated the steps involved in the feature extraction, dimensionality reduction, learning and expectation processes and we have compared these steps quantitatively using a corpus of percussive excerpts. Three expectation schemes, namely independent, joint, and when|what schemes were defined to take into account alternative viewpoints regarding how musical events are organized. From a cognitive perspective those schemes may be associated to distinct processing pathways of the musical stream (Peretz and Zatorre, 2005).

A set of statistics has been presented to characterize the system learning abilities. An evaluation of the system components has been performed. The results show that, on the one hand, the system's ability to produce an expectation timeline that agrees with ground truth annotations depends on the quality of the transcription process, and that the performance of the when|what scheme is more severely degraded when the transcription accuracy degrades. On the other hand, even if the representation of the system does not accurately fit the ground truth, our results suggested that the sys-

tem is able to use and maintain an internal representation of the musical stream to identify patterns whose regularities can be learned. In this latter case, the transcription step may be seen as a musically-informed sparse representation process. Therefore, we suggest that the model may be evaluated according to the usefulness and robustness of its representation rather than using ground truth annotations. Besides, we have suggested that the expectation signal may describe the structure of musical patterns in a causal manner and have shown the expectation entropy measure could provide information about the structure of musical streams. This expectation-based signals may provide an alternative approach to perform structural segmentation tasks.

Integrating representation and expectation models into musical systems By their nature, causal systems accommodate with the time dynamics of the sensory data they process. In the case of audio, real-time systems can be implemented to provide a seamless interaction with users, and enables novel uses for musical or educational purposes, as well as entertainment. Among the possible applications of this work are: musical learning applications that do not require a specific hardware such as a keyboard, accompaniment applications, or audio-driven games. Here, our main motivation is to demonstrate that even if there is room for improvement in almost all components of the system, it is already possible to design novel interactive settings and applications that take advantage of representing and creating expectations from audio signals. Because musicians and enthusiasts are among the most valuable users of the technologies that may derive from this research, they should be given ways to access and understand those technologies. This way, users may help to both broaden and refine the spectrum of possible systems that enable a meaningful musical interaction.

8.2 Open issues

In spite of the contributions and achievements presented here, we would like to see this research as starting point which highlights interesting issues in the field of expectation modelling. Here we attempt to describe the most relevant issues and work directions to this work.

8.2.1 Measuring mismatch with the environment

One objection that can be made concerning the approach proposed here is that it creates and maintains a representation of the musical events, generates expectations based on this, but does not retrospectively compare these expectations with actual events to refine its representation. This would correspond to Ellis' reconciliation process (Ellis, 1996) or the appraisal step in the theory developed by Huron (2006). To this extent, even though our system can create symbolic expectations that can be mapped into absolute time by the representation layer, there is no further feedback processing nor reuse of the expectation mismatch with the environment. It is worth noting

that our architecture could handle this additional processing, but we need to properly define mismatch measurements to track which expectations are fulfilled, and *how* they are fulfilled. Some starting points to this would be the distances measures proposed by Paiement (2008), as well as works focusing on audio similarity (Pampalk et al., 2008).

8.2.2 Scales of music processing

Another outstanding issue in our research is the need to address the various temporal scales involved in music processing and learning.

Representation over multiple time scales In our model, we have focused in representing musical structure in terms of relations between event onsets over several acoustic dimensions. While being simple enough to be computationally tractable and rich enough to describe several aspects of musical sequences at the same time, this time scale may be inaccurate to model more precise acoustic inflections that could contain relevant information. Conversely, if we aim consider a musical excerpt from a broader viewpoint, we might need to use a longer time scale. It is worth noting that, by creating an expectation signal between event onsets (that can be considered in the timescale of a note), the system may able to implicitly perform segmentation on a bar timescale, as the expectation entropy patterns suggest (see Section 6.7.7). As a future work, it may be possible to represent and learn the regularities of the expectation entropy patterns.

Learning timescale In the majority of the experiments presented here, we have repeatedly presented to the system short music excerpts and assessed whether the system was able to follow those excerpts through repetitions. However, because the model's running state proceeds in a causal way, our implementation could virtually analyze musical streams of arbitrary duration. Consequently, we need to investigate how to assess longer-term effects of learning, what musical memory contents are maintained and how they are maintained through the exposure to successive and possibly distinct musical sequences. For instance, we would like to assess to which extent our model can keep some information about previous excerpts when heading at the end of a playlist. One actual limitation to progress in that direction is that our model performs the bootstrap estimation only at the beginning of any exposure. Alternative strategies should be investigated, such as performing bootstrap when a new representation is needed, or continuously performing the bootstrap step in parallel to the running state.

8.2.3 Towards an account of auditory learning experiments using the what/when model

In Chapter 4 we have proposed a framework to modelling the acquisition of statistical regularities in tone sequences, represented in a symbolic form. Then, in Chapter 6, we have introduced an expectation model that learns to represent auditory events in a meaningful way. From these two contributions,

it seems natural to investigate whether the what/when expectation model can provide an account of empirical results dealing with auditory learning experiments that focus on the interaction between acoustical cues and time structures (Newport and Aslin, 2004; Tillmann and McAdams, 2004), but this would be better included in the research agenda of music cognition scientists.

8.3 A personal concluding note

This dissertation is in essence a multidisciplinary account to model music expectation from areas such as auditory perception, learning, expectation modelling, and information theory. As such, an important part of my research has consisted in building bridges across these disciplines, and trying to view issues that were specific of a given area from a distinct viewpoint. Firstly, this dissertation aims to provide tools to cognitive science and sequence learning researchers to address computationally real-world auditory and musical stimuli. Likewise, it aims at enabling MIR and auditory perception researchers to investigate the effects of learning and the usefulness of an ever-evolving representation of the musical signal. Finally, I hope that the contributions presented here can serve as a basis for designing an interactive music system driven by cognitive principles, pattern recognition and predictive learning.

Bibliography

Abdallah, S. (2008). A critique of dubnov's "Information Rate". Technical Report C4DM-TR08-11, Center for Digital Music, Queen Mary, University of London.

Abdallah, S. and Plumbley, M. (2009). Information dynamics: patterns of expectation and surprise in the perception of music. *Connection Science*, 21(2):89–117.

Abdallah, S. A. (2002). *Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*. PhD thesis, Kingś College London, London, UK.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19:716–723.

Allan, L. (1979). The perception of time. *Perception and Psychophysics*, 26:340–354.

Allauzen, C., Crochemore, M., and Raffinot, M. (1999). Factor oracle: A new structure for pattern matching. In *Proceedings of the 26th Conference on Current Trends in Theory and Practice of Informatics on Theory and Practice of Informatics*, pages 295 – 310.

Altmann, G. T. M., Dienes, Z., and Goode, A. (1995). Modality independence of implicitly learned grammatical knowledge. *Journal of Experimental Psychology-Learning Memory and Cognition*, 21(4):899–912.

Assayag, G., Bloch, G., and Chemillier, M. (2006). «OMax-Ofon». In *Proceedings of Sound and Music Computing (SMC)*, Marseille, France.

Assayag, G. and Dubnov, S. (2004). Using Factor Oracles for machine Improvisation. *Soft Computing*, 8(9):604–610.

Aucouturier, J. J. and Sandler, M. (2001). Segmentation of musical signals using hidden markov models. *Preprints Audio Engineering Society*.

Barlow, H. (2001). The exploitation of regularities in the environment by the brain. *Behavioral and Brain Sciences*, 24:602–607.

Berkes, P. and Zito, T. (2007). Modular toolkit for data processing (version 2.1).

Bharucha, J. J. and Todd, P. M. (1989). Modeling the perception of tonal structure with neural nets. *Computer Music Journal*, 13:44–53.

Bigand, E., Perruchet, P., and Boyer, M. (1998a). Implicit learning of an artificial grammar of musical timbres. *Cahiers de psychologie cognitive*, 17(3):577–600.

Bigand, E., Perruchet, P., and Boyer, M. (1998b). Implicit learning of an artificial grammar of musical timbres. *Cahiers de psychologie cognitive*, 17(3):577–600.

Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Boltz, M. G. (1993). The generation of temporal and melodic expectancies during musical listening. *Perception and Psychophysics*, 53:585–585.

Bottou, L. (2004). Stochastic learning. *Advanced Lectures On Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003: Revised Lectures*.

Botvinick, M. M. and Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, 113(2):201–233.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1976). *Time series analysis: forecasting and control*. Holden-day San Francisco.

Bregman, A. S. (1990). *Auditory Scene Analysis*. MIT Press, Cambridge, MA.

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

Brossier, P., Bello, J., and Plumbley, M. (2004). Fast labelling of notes in music signals. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, pages 331–336, Barcelona, Spain.

Brown, G. J. (1993). Computational auditory scene analysis: A representational approach. *The Journal of the Acoustical Society of America*, 94(4):2454.

Burton, A. R. and Vladimirova, T. (1999). Generation of musical sequences with genetic techniques. *Computer Music Journal*, 23(4):59–73.

Caclin, A., Brattico, E., Tervaniemi, M., Näätänen, R., Morlet, D., Giard, M., and McAdams, S. (2006). Separate neural processing of timbre dimensions in auditory sensory memory. *Journal of Cognitive Neuroscience*, 18(12):1959–1972.

Cano, P. (2006). *Content-Based Audio Search: From Fingerprinting To Semantic Audio Retrieval*. PhD thesis, Universitat Pompeu Fabra.

Carpinteiro, O. (2000). A hierarchical self-organising map model for sequence recognition. *Pattern Analysis and Applications*, 3:279–287.

Casey, M. (2002). General sound classification and similarity in MPEG-7. *Organised Sound*, 6(02):153–164.

Cemgil, A. T., Kappen, H. J., Dietterich, T. G., and Becker, S. (2002). Tempo tracking and rhythm quantization by sequential monte carlo. In *Dietterich, Th. G.; Becker, S.; Ghahramani, Z.(eds.), Advances in neural information processing systems 14: proceedings of the 2002 conference*, page 1361. Cambridge, Mass.; London, England: MIT Press.

Clarke, E. (1989). The perception of expressive timing in music. *Psychological Research*, 51:2–9.

Clarke, E. F. (1987). Levels of structure in the organization of musical time. *Contemporary music review*, 2(1):211–238.

Cleary, J. G. and Witten, I. H. (1984). A comparison of enumerative and adaptive codes. *IEEE Transactions on Information Theory*, 30(2):306–315.

Cleeremans, A. (1993). *Mechanisms of implicit learning: Connectionist models of sequence processing*. MIT press, Cambridge, MA.

Cleeremans, A. and McClelland, J. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120(3):235–253.

Collins, N. (2004). On onsets on-the-fly: Real-time events segmentation and categorization as a compositional effect. In *Proceedings of the First Sound and Music Computing Conference (SMC'04)*, Paris, France.

Conklin, D. and Anagnostopoulou, C. (2001). Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference*, pages 479–485, La Havana, Cuba.

Cont, A. (2008). *Modeling Musical Anticipation: From the time of music to the music of time*. PhD thesis, University of Paris 6 and University of California in San Diego.

Cooke, M. (1993). *Modelling auditory processing and organisation*. Cambridge University Press, Cambridge, UK.

Cooper, G. and Meyer, L. (1960). *The rhythmic structure of music*. University of Chicago Press, Chicago, IL.

Cournapeau, D. (2006). Pyem, a python package for gaussian mixture models. Technical report, University of Kyoto, Graduate School of Informatics.

Creel, S. C., Newport, E. L., and Aslin, R. N. (2004). Distant melodies: Statistical learning of nonadjacent dependencies in tone sequences. *Learning, Memory*, 30:1119–1130.

Dalla Bella, S., Peretz, I., and Aronoff, N. (2003). Time course of melody recognition: a gating paradigm study. *Perception and Psychophysics*, 65(7).

Davies, M. E. P., Brossier, P. M., and Plumbley, M. D. (2005). Beat tracking towards automatic musical accompaniment. In *Proceedings of the 118th AES convention*, Barcelona, Spain.

Davies, M. E. P. and Plumbley, M. D. (2005). Beat tracking with a two state model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, pages 241–244, Philadelphia, Penn., USA.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Pattern Recognition and Artificial Intelligence, RCH Chen, ed., Academic Press: New York*, 28(4):357–366.

Demany, L., McKenzie, B., and Vurpillot, E. (1977). Rhythm perception in early infancy. *Nature*, 266:718–719.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society*, B-39:1–38.

Desain, P. and Honing, H. (2003). The formation of rhythmic categories and metric priming. *Perception*, 32:341–365.

Dixon, S. (1997). Beat Induction and Rhythm Recognition. In *Proceedings of the 10th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence*, page 320. Springer-Verlag.

Downie, J. S. (2003). Music information retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340.

Drake, C. (1998). Psychological processes involved in the temporal organization of complex auditory sequences: universal and acquired processes. *Music Percept.*, 16:11–26.

Dubnov, S. (2006). Spectral anticipations. *Computer Music Journal*, 30(2):63–83.

Dubnov, S. (2008). Unified view of prediction and repetition structure in audio signals with application to interest point detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):327–337.

Duxbury, C., Bello, J., Davies, M., and Sandler, M. (2003a). Complex domain onset detection for musical signals. *Proceedings Digital Audio Effects Workshop (DAFx)*.

Duxbury, C., Bello, J. P., Davies, M., and Sandler, M. (2003b). Complex domain onset detection for musical signals. In *Proc. Digital Audio Effects Workshop (DAFx)*.

Eck, D. and Schmidhuber, J. (2002). Learning the long-term structure of the blues. *Lecture Notes in Computer Science, Proceedings of ICANN Conference*, 2415:284–289.

Ellis, D. P. W. (1996). *Prediction-driven Computational Auditory Scene Analysis*. PhD thesis, Massachusetts Institute of Technology, MA, USA.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Ferrand, M., Nelson, P., and Wiggins, G. (2002). A probabilistic model for melody segmentation. In *2nd International Conference on Music and Artificial Intelligence*, University of Edinburgh, UK.

Foote, J. (1999). Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80. ACM New York, NY, USA.

Franklin, J. A. (2004). Recurrent neural networks and pitch representations for music tasks. In Barr, V. and Markov, Z., editors, *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*. AAAI Press.

Ghezzi, C. and Mandrioli, D. (1979). Incremental parsing. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 1(1):58–70.

Gillet, O. and Richard, G. (2004). Automatic transcription of drum loops. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, 4:269–272.

Gillet, O. and Richard, G. (2006). Enst-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR 2006)*, pages 156–159.

Gómez, E. (2006). *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171.

Goto, M. (2003). A chorus-section detecting method for musical audio signals. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 5.

Goto, M. and Muraoka, Y. (1995). A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference*, pages 171–174.

Gouyon, F. and Dixon, S. (2005). A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54.

Gouyon, F. and Herrera, P. (2003). A beat induction method for musical audio signals. In *Digital media processing for multimedia interactive services: proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services: Queen Mary, University of London, 9-11 April 2003*, page 281. World Scientific Pub Co Inc.

Gouyon, F., Pachet, F., and Delerue, O. (2000). On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*. Citeseer.

Grachten, M., Arcos, J., and López de Mantaras, R. (2006). A case based approach to expressivity-aware tempo transformation. *Machine Learning*, 65(2-3):411–437.

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

Griffith, N. and Todd, P. M. (1999). *Musical Networks. Parallel Distributed Perception and Performance*. The MIT Press, Cambridge, MA.

Hainsworth, S. (2006). Beat tracking and musical metre analysis. In *Signal processing methods for music transcription*, page 440. Springer-Verlag.

Hannon, E. E. and Trehub, S. E. (2005). Metrical categories in infancy and adulthood. *Psychological Science*, 16(1):48–55.

Hasty, C. (1997). *Meter as Rhythm*. Oxford University Press.

Hazan, A. (2005a). Billaboop: Real-time voice-driven drum generator. In *Proceedings of Audio Engineering Society, 118th Convention*.

Hazan, A. (2005b). Performing expressive rhythms with billaboop voice-driven drum generator. In *Proceedings of 8th Intl. Conference on Digital Audio Effects*, Madrid, Spain.

Hazan, A. (2005c). Towards automatic transcription of oral expressive performances. In *Proceedings of the Intelligent User Interfaces Conference (IUI 2005)*.

Hazan, A., Brossier, P., Holonowicz, P., Herrera, P., and Purwins, H. (2007). Expectation along the beat: A use case for music expectation models. In *Proceedings of International Computer Music Conference 2007*, pages 228–236, Copenhagen, Denmark.

Hazan, A., Brossier, P., Marxer, R., and Purwins, H. (2008a). What/when causal expectation modelling applied to percussive audio. In *The Journal of the Acoustical Society of America*, volume 123, page 3800.

Hazan, A., Holonowicz, P., Salselas, I., Herrera, P., Purwins, H.;Knast, A., and Durrant, S. (2008b). Modeling the acquisition of statistical regularities in tone sequences. In *Prceedings of the 30th Annual Meeting of the Cognitive Science Society*, pages 1088–1094.

Hazan, A., Marxer, R. ;Brossier, P., Purwins, H.;Herrera, P., and Serra, X. (2009). What/when causal expectation modelling applied to audio signals. *Connection Science*, 21:119–143.

Hazan, A., Ramirez, R., Maestre, E., Perez, A., and Pertusa, A. (2006). Modelling expressive performance: A regression tree approach based on strongly typed genetic programming. In *Proceedings of the European Workshop on Evolutionary Music and Art*, volume 3907, pages 676–672, Lausanne, Switzerland. Springer.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738.

Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589.

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

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Homer, A. and Goldberg, D. E. (1991). Genetic algorithms and Computer-Assisted music composition. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, volume 51, page 437, San Fransisco, CA. Morgan Kaufmann Publishers.

Huron, D. (2006). *Sweet anticipation*. MIT Press, Cambridge, MA.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. *Advances in Neural Information Processing Systems*, 15:593–600.

Jayant, N. S. and Noll, P. (1984). *Digital coding of waveforms*. Prentice-Hall, Englewood Cliffs, NJ.

Jehan, T. (2005). *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, MA, USA.

Jones, M. R. and Boltz, M. (1989). Dynamic attending and responses to time. *Psychological Review*, 96(3):459–491.

Kapur, A., Benning, M., and Tzanetakis, G. (2004). Query by beat boxing: Music retrieval for the dj. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*.

Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, USA.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3):1–6.

Krumhansl, C. L. (1979). The psychological representation of musical pitch in a tonal context. *Cognitive Psychology*, 11(3):346–74.

Krumhansl, C. L. (1989). Why is timbre so hard to understand. In Nielzén, S. and Olsson, O., editors, *Structure and Perception of electroacoustic sound and music*, pages 43–53. Excerpta Medica, Amsterdam.

Kuhn, G. and Dienes, Z. (2008). Learning of non local dependencies. *Cognition*, 106(1):184–206.

Lacoste, A. and Eck, D. (2007). A supervised classification algorithm for note onset detection. *EURASIP J. Appl. Signal Process.*, 2007(1):153–153.

Ladefoged, P. (1989). A note on Information conveyed by vowels. *The Journal of the Acoustical Society of America*, 85:2223.

Large, E. W. (2000). Rhythm categorization in context. In *Proceedings of the International Conference on Music Perception and Cognition*.

Lartillot, O., Dubnov, S., Assayag, G., and Bejerano, G. (2001). Automatic modeling of musical style. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, La Havana, Cuba.

Lashley, K. S. and Jeffress, L. A. (1951). Cerebral mechanisms in behavior. *The problem of serial order in behavior*, pages 112–136.

Lerdahl, F. and Jackendoff, R. A. (1983). *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA.

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nat Neurosci*, 5(4):356–363.

London, J. (2002). Cognitive constraints on metric systems: Some observations and hypotheses. *Music Perception*, 19(4):529–550.

London, J. (2006). How to talk about musical meter. Colloquium talks, Carleton College, MN.

Loui, P., Wessel, D., and Hudson Kam, C. (2006). Adquiring new musical grammars: a statistical learning approach. In *Proceedings of the International Conference on Music Perception and Cognition*, Bolgna, Italy.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Publisher: Cambridge University Press, Cambridge, UK. Available from http://www.inference.phy.cam.ac.uk/mackay/itila/.

Maess, B., Koelsch, S., Gunter, T. C., and Friederici, A. D. (2001). Musical syntax is processed in Broca's area: an MEG study. *Nature Neuroscience*.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580.

Markov, A. A. (1971). *Theory of algorithms*. Israel Program for Scientific Translations.

Marxer, R., Holonowicz, P., Purwins, H., and Hazan, A. (2007). Dynamical hierarchical self-organization of harmonic, motivic, and pitch categories. In *Proceedings of the Music, Brain and Cognition Workshop, part of the NIPS conference*, Vancouver, Canada.

Marxer, R., Purwins, H., and Hazan, A. (2008). An f-measure for evaluation of unsupervised clustering with non-determined number of clusters. Technical report, Universitat Pompeu Fabra, Music Technology Group.

Masri, P. (1996). *Computer modeling of Sound for Transformation and Synthesis of Musical Signal*. PhD thesis, University of Bristol, Bristol, U.K.

Mathews, M. V., Roberts, L. A., and Pierce, J. R. (1984). Four new scales based on nonsuccessive-integer-ratio chords. *The Journal of the Acoustical Society of America*, 75(S1):S10.

McAdams, S. and Bregman, A. (1979). Hearing musical streams. *Computer Music Journal*, 3(4):26–60.

McAdams, S., Winsberg, S., Donnadieu, S., Soete, G., and Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58(3):177–192.

Meyer, L. B. (1956). *Emotion and meaning in music*. University of Chicago Press, Chicago, IL.

Moffat, A. (1990). Implementing the PPM data compression scheme. *IEEETCOMM: IEEE Transactions on Communications*, 38:1917–1921.

Mozer, M. (1994). Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing. *Connection Science*, 6:247–280.

Narmour, E. (1990). *The analysis and cognition of basic melodic structures: the Implication-Realization model*. University of Chicago Press, Chicago, IL.

Narmour, E. (1992). *The analysis and cognition of melodic complexity: the Implication-Realization model*. University of Chicago Press, Chicago, IL.

Newport, E. L. and Aslin, R. N. (2004). Learning at a distance: I. statistical learning of Non-Adjacent dependencies. *Cognitive Psychology*, 48(2):127–162.

Ong, B. S. (2006). *Structural analysis and segmentation of music signals*. PhD thesis, Universitat Pompeu Fabra.

O'Reilly, R. C. and Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain.* MIT Press.

Pachet, F. (2003). The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341.

Paiement, J. F. (2008). *Probabilistic Models for Music.* PhD thesis, Ecole Polytechnique Féderale De Lausanne.

Palmer, C. (1989). Mapping musical thought to musical performance. *J. of Experimental Psychology - Human Perception and Performance*, 15(12):331–346.

Palmer, C. and Krumhansl, C. L. (1990). Mental representations for musical meter. *J. of Experimental Psychology - Human Perception and Performance*, 16(4):728–741.

Pampalk, E., Herrera, P., and Goto, M. (2008). Computational models of similarity for drum samples. *IEEE Transactions on Audio Speech and Language Processing*, 16(2):408.

Paulus, J. and Klapuri, A. (2003). Model-based event labeling in the transcription of percussive audio signals. In Davies, M., editor, *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, pages 73–77, London, UK.

Pearce, M. T. and Wiggins, G. A. (2004). Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Project Report.*

Peeters, G., Burthe, A. L., and Rodet, X. (2002). Toward automatic music audio summary generation from signal analysis. In *Proc. International Conference on Music Information Retrieval*, pages 94–100. Citeseer.

Peretz, I. and Zatorre, R. (2005). Brain organization for music processing. *Annual Review of Psychology*, 56(1):89–114.

Povel, D. J. and Essens, P. (1985). Perception of temporal patterns. *Music Perception*, 2(4):411–440.

Purwins, H., Grachten, M., Herrera, P., Hazan, A., Marxer, R., and Serra, X. (2008). Computational models of music perception and cognition II: Domain-specific music processing. *Physics of Life Reviews*, 5:169–182.

Ramirez, R. and Hazan, A. (2005). Understanding expressive music performance using genetic algorithms. In *Proceedings of the European Workshop on Evolutionary Music and Art*, pages 508–516. Springer.

Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning & Verbal Behavior. Vol*, 6(6):855–863.

Roads, C. (2004). *Microsound*. MIT Press, Cambridge, MA.

Rothstein, W. (1981). *Rhythm and the Theory of Structural Levels*. PhD thesis, Yale University.

Rumelhart, D. and McLelland, J. (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA.

Saffran, J., Johnson, E., Aslin, R., and Newport, E. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, 70(1):27–52.

Saffran, J. R., Newport, E. L., and Aslin, R. N. (1996). Word segmentation: the role of distributional cues. *Journal of Memory and Language*, 35:606–621.

Schaeffer, P. (1966). *Traité des objets musicaux*. Le Seuil, Paris.

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103:588.

Schellenberg, E. G. (1996). Expectancy in melody: Tests of the implication-realization model. *Cognition*, 58:75–125.

Schouten, J. F. (1968). The perception of timbre. In *Reports of the 6th International Congress on Acoustics*.

Schwarz, D. (2004). *Data-Driven Concatenative Sound Synthesis*. PhD thesis, Ircam - Centre Pompidou, Paris, France.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Smith, L. (1996). Listening to musical rhythms with progressive wavelets. In *TENCON '96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications*, volume 2, pages 508–513 vol.2.

Snyder, B. (2000). *Music and Memory: An Introduction*. The MIT Press, Cambridge, MA.

Sridharan, D., Levitin, D., Chafe, C., Berger, J., and V., M. (2007). Neural dynamics of event segmentation in music: converging evidence for dissociable ventral and dorsal networks. *Neuron*, 55(3):521–532.

Tanghe, K., Degroeve, S., and Baets, B. D. (2005a). An algorithm for detecting and labeling drum events in polyphonic music. In *Mirex Drum Recognition Contest (part of International Symposium on Music Information Retreival)*.

Tanghe, K., Lesaffre, M., Degroeve, S., Leman, M., Baets, B. D., and Martens, J. P. (2005b). Collecting ground truth annotations for drum detection in polyphonic music. In *Proc. 6th Int. Conf. on Music Information Retrieval (ISMIR 2005)*, pages 50–57.

Terhardt, E. (1998). *Akustische Kommunikation*. Springer, Berlin.

Tillmann, B. (2008). Music cognition: Learning, perception, expectations. In Kronland-Martinet, R., Ystad, S., and Jensen, K., editors, *Computer Music Modeling and Retrieval. Sense of Sounds*, pages 11–33. Springer, Berlin.

Tillmann, B., Bharucha, J. J., and Bigand, E. (2000). Implicit learning of tonality: A self-organizing approach. *Psychological Review*, 107(4):885–913.

Tillmann, B. and McAdams, S. (2004). implicit learning of musical timbre sequences: statistical regularities confronted with acoustical (dis)similarities. *Journal of experimental psychology, learning, memory and cognition*, 30(5):1131–1142.

Todd, P. M. and Loy, D. G. (1991). *Music and Connectionism*. MIT Press.

Wang, A. (2003). An industrial strength audio search algorithm. In *International Conference on Music Information Retrieval (ISMIR 2003)*, pages 7–13.

Wang, Y., Tang, J., Ahmaniemi, A., Vaalgamaa, M., Center, N. R., and Tampere, F. (2003). Parametric vector quantization for coding percussive sounds in music. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3.

Warren, R. and Warren, R. (1970). Auditory illusions and confusions. *Scientific American*, 223(12):30–36.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, Cambridge, MA.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wessel, D. L. (1979). Timbre space as a musical control structure. *Computer music journal*, pages 45–52.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Yoshii, K., Goto, M., and Okuno, H. G. (2005). Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates. In *Mirex Drum Recognition Contest (part of International Symposium on Music Information Retreival)*.

Appendix A: Full list of publications

Journal Articles

Hazan, A., Marxer, R. ;Brossier, P., Purwins, H.;Herrera, P., and Serra, X. (2009). What/when causal expectation modelling applied to audio signals. *Connection Science*, 21:119–143.

Coath, M., Denham, S. L., Smith, L. M., Honing, H., Hazan, A., Holonowicz, P., and Purwins, H. (2009). Model cortical responses for the detection of perceptual onsets and beat tracking in singing. *Connection Science*, 21:193–205.

Purwins, H., Herrera, P., Grachten, M., Hazan, A., Marxer, R., and Serra, X. (2008). Computational models of music perception and cognition I: The perceptual and cognitive processing chain. *Physics of Life Reviews*, 5:151–168.

Purwins, H., Grachten, M., Herrera, P., Hazan, A., Marxer, R., and Serra, X. (2008). Computational models of music perception and cognition II: Domain-specific music processing. *Physics of Life Reviews*, 5:169–182.

Ramirez, R., Hazan, A., Maestre, E., and Serra, X. (2008). A genetic rule-based expressive performance model for jazz saxophone. *Computer Music Journal*, 32:38–50.

Ramirez, R. and Hazan, A. (2006). A tool for generating and explaining expressive music performances of monophonic jazz melodies. *International Journal on Artificial Intelligence Tools*, 15:673–691.

Ramirez, R., Hazan, A., Gómez, E., Maestre, E., and Serra, X. (2005). Discovering expressive transformation rules from saxophone jazz performances. *Journal of New Music Research*, 34:319–330.

Book Chapters

Ramirez, R., Hazan, A., Maestre, E., and Serra, X. (2006). A machine learning approach to expressive performance in jazz standards. In Petrushin, V. A. and Khan, L., editors, *Multimedia Data Mining and Knowledge Discovery*. Springer.

Dissertations

Hazan, A. (2006). Computational modeling of expressive music performance new machine learning approaches for dealing with real-world data. *Doctoral Pre-Thesis Work. UPF.*

Hazan, A. (2004). Interfaz oral para el reconocimiento de ritmos. *Bachelor Thesis, Polytechnical University of Catalunya.*

Conference Proceedings

Hazan, A., Holonowicz, P., Salselas, I., Herrera, P., Purwins, H.;Knast, A., and Durrant, S. (2008). Modeling the acquisition of statistical regularities in tone sequences. In *Prceedings of the 30th Annual Meeting of the Cognitive Science Society*, pages 1088–1094.

Hazan, A., Brossier, P., Holonowicz, P., Herrera, P., and Purwins, H. (2007). Expectation along the beat: A use case for music expectation models. In *Proceedings of International Computer Music Conference 2007*, pages 228–236, Copenhagen, Denmark.

Hazan, A., Brossier, P., Marxer, R., and Purwins, H. (2007). What/when causal expectation modelling in monophonic pitched and percussive audio. In *Music, Brain and Cognition Workshop, part of the Neural Information Processing Conference*, Vancouver, Canada.

Hazan, A. and Ramirez, R. (2006). Modelling expressive music performance using evolutionary regression trees. In *Proceedings of the Evolutionary Computation Workshop, ECCAI conference*, Trento, Italia.

Hazan, A., Grachten, M., and Ramirez, R. (2006). Evolving performance models by performance similarity beyond note-to-note transformations. In *Proceedings of the International Symposium on Music Information Retrieval*, Victoria, Canada.

Hazan, A., Ramirez, R., Maestre, E., Perez, A., and Pertusa, A. (2006). Modelling expressive performance: A regression tree approach based on strongly typed genetic programming. In *Proceedings of the European Workshop on Evolutionary Music and Art*, volume 3907, pages 676–672, Lausanne, Switzerland. Springer.

Maestre, E., Hazan, A., Ramirez, R., and Perez, A. (2006). Using concatenative synthesis for expressive performance in jazz saxophone. In *Proceedings of the International Computer Music Conference*, New Orleans, US.

Ramirez, R., Hazan, A., and Maestre, E. (2006). A sequential covering evolutionary algorithm for expressive music performance. In *Conference on Innovative Applications of Artificial Intelligence*, Boston, USA. AAAI Press.

Ramirez, R. and Hazan, A. (2005). Understanding expressive music performance using genetic algorithms. In *Proceedings of the European Workshop on Evolutionary Music and Art*, pages 508–516. Springer.

Hazan, A. (2005). Billaboop: Real-time voice-driven drum generator. In *Proceedings of Audio Engineering Society, 118th Convention.*

Hazan, A. (2005). Performing expressive rhythms with billaboop voice-driven drum generator. In *Proceedings of 8th Intl. Conference on Digital Audio Effects*, Madrid, Spain.

Hazan, A. (2005). Towards automatic transcription of oral expressive performances. In *Proceedings of the Intelligent User Interfaces Conference (IUI 2005)*.

Ramirez, R., Hazan, A., and Maestre, E. (2005). Intra-note features prediction model for jazz saxophone performance. In *International Computer Music Conference*, Barcelona, Spain.

Ramirez, R., Hazan, A., Gómez, E., and Maestre, E. (2004). A machine learning approach to expressive performance in jazz standards. In *Proceedings of 10th International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA.

Ramirez, R., Hazan, A., Gómez, E., and Maestre, E. (2004). Understanding expressive transformations in saxophone jazz performances using inductive machine learning. In *Proceedings of Sound and Music Computing '04*, Paris, France.

Appendix B: Sound Examples

This appendix contains a list [1] of sound examples that illustrate the process of sonifying the expectation of audio excerpts using the Billabio software, presented in Chapter 7. For each excerpt, we provide the original sound (left column), the sonification of the expectation immediately after bootstrap (middle column), and the sonification of the expectation after the system has processed a few repetitions of the original sound (right column).

Type of sound	*Original excerpt*	*Beginning*	*End*
Drum'n bass	Audio.1	Audio.2	Audio.3
Drum 1	Audio.4	Audio.5	Audio.6
Drum 2	Audio.7	Audio.8	Audio.9
Voice	Audio.10	-	Audio.11
Keyboard	Audio.12	Audio.13	Audio.14

1. This list can also be accessed using the following link `http://www.dtic.upf.edu/~ahazan/thesis/examples.html`

This thesis has been written using LaTeX