Bayesian Methods in Music Modelling

Paul Halliday Peeling
Clare College

December 19, 2010

Declaration

This dissertation is submitted for the degree of Doctor of Philosophy. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. This dissertation does not exceed 50,000 words and 50 figures.
Paul H. Peeling

Abstract

This thesis presents several hierarchical generative Bayesian models of musical signals designed to improve the accuracy of existing multiple pitch detection systems and other musical signal processing applications whilst remaining feasible for real-time computation. At the lowest level the signal is modelled as a set of overlapping sinusoidal basis functions. The parameters of these basis functions are built into a prior framework based on principles known from musical theory and the physics of musical instruments. The model of a musical note optionally includes phenomena such as frequency and amplitude modulations, damping, volume, timbre and inharmonicity. The occurrence of note onsets in a performance of a piece of music is controlled by an underlying tempo process and the alignment of the timings to the underlying score of the music.

A variety of applications are presented for these models under differing inference constraints. Where full Bayesian inference is possible, reversible-jump Markov Chain Monte Carlo is employed to estimate the number of notes and partial frequency components in each frame of music. We also use approximate techniques such as model selection criteria and variational Bayes methods for inference in situations where computation time is limited or the amount of data to be processed is large. For the higher level score parameters, greedy search and conditional modes algorithms are found to be sufficiently accurate.

We emphasize the links between the models and inference algorithms developed in this thesis with that in existing and parallel work, and demonstrate the effects of making modifications to these models both theoretically and by means of experimental results.

Acknowledgments

First of all I would like to thank my supervisor Prof. Simon Godsill, whose insight and direction has guided much of the work in this thesis, and attention to detail has been invaluable.

Very special thanks to Dr. Taylan Cemgil, for being a constant source of inspiration and encouragement on this project and for providing so many examples and resources which made this possible. I am very grateful to all those who I also have collaborated with at various points over the last four years: Drs. Sumeetpal Singh, Nick Whiteley, Daniel Clark, Onur Dikmen and Chung-fai Li.

Thanks to all the staff and students in the Signal Processing Laboratory, especially Henry, Jonathan and Simon for on- and off-topic coffee room discussions, and to Janet and Rachel for putting in so much effort to support us all. Thanks also to the folks at Featurespace, especially Bill, Dave and Kirsty, for encouraging me along the way.

My final thanks go to my family and friends, who haven't needed to understand what this thesis is about to guide and support me: Matthew, Glen, Denise, Glen, Elliot, Theodore, Bob, Jon, Ding, Yuwei, Ted, Olly,

Thomas. Thanks to Mum, Dad and Katie for their love and support throughout the university years. Most of all, thanks to Angel for caring and loving every time. This thesis is dedicated to our new arrival.

Notation

$y \sim p(y)$	y is sampled from the probability distribution $p(y)$
$p(y\|\theta)$	The conditional probability density of y given θ
$\mathcal{N}(y; \mu, \sigma^2)$	y is normally distributed with mean μ and variance σ^2
$\mathcal{N}_C(y; \mu, \sigma^2)$	Complex normal distribution
$\mathbf{A}_{m,n}$	The element of matrix $\mathbf{A}$ in the mth row and nth column
$E_{p(y)}[f(y)]$	The expectation of the function $f(y)$ under the probability distribution $p(y)$
$\langle f(y) \rangle_{p(y)}$	The expectation of the function $f(y)$ under the probability distribution $p(y)$
f_0	The fundamental frequency of a musical note
Tr $\mathbf{A}$	The trace of matrix $\mathbf{A}$
$\mathbf{A}^\dagger$	Pseudo-inverse of matrix $\mathbf{A}$
A440	The note A, which has a pitch of 440 Hz
$\hat{\theta}$	A point estimate of the parameters θ
$y_{1:K}$	The set of observations $\{y_1, \ldots, y_K\}$

Acronyms

DFT	Discrete Fourier transform
STFT	Short-time Fourier transform (spectrogram)
(M)DCT	(Modified) Discrete Cosine transform
MIDI	Musical instrument Digital Interface
MIREX	Music Information Retrieval Exchange
NMF	Non-negative Matrix Factorization
HMM	Hidden Markov Model
ML	Maximum-likelihood parameter estimate
MAP	Maximum *a posteriori* parameter estimate
EM	Expectation-maximization algorithm
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
GMM	Gaussian mixture model
ACF	Autocorrelation function
SNR	Signal-noise ratio

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Background

This thesis is concerned with Bayesian methods for the modelling of musical signals. Musical signals are rich in structure, and are capable of evoking emotional and aesthetic response in the listener. When processing music, much of this structure is known in advance. Thus we may use Bayesian methods to infer some of the unknown aspects of the musical signal in a signal processing setting.

Bayesian methods center around the application of Bayes' rule to statistical models of observed data y. The unknown information about the musical signal y is encapsulated in a set of parameters θ. We define a statistical model $p(y|\theta)$, the *likelihood*, which describes how the signal is related to these parameters. To complete the picture, we define a *prior* $p(\theta)$ which describes what we know about the parameters before we observe any data. A Bayesian model thus may be represented by its joint probability distribution

$$p(y, \theta) = p(y|\theta)\, p(\theta)$$

Using the prior and the likelihood, the *posterior* distribution $p(\theta|y)$ of the parameters after we observe the signal is given by Bayes' rule

$$p(\theta|y) \propto p(y|\theta)\, p(\theta) = \frac{p(y|\theta)\, p(\theta)}{p(y)} \tag{1.1}$$

The quantity $p(y)$ in (1.1) is both the normalization constant of the posterior $p(\theta|y)$ and also the *marginal* distribution of the observed data for our particular choice of model $p(y, \theta)$.

$$p(y) = \int p(y, \theta)\ \mathrm{d}\theta$$

$p(y)$ is therefore known as the marginal likelihood or the *evidence* of the data, and may be used to evaluate which model from a range of potential models best explains the data observed.

From a musical signal processing perspective, the unknown parameters θ represent a hierarchy of musical information. At the highest level we have the cognitive concepts of genre, mood, style and so forth, which

may be used to group different pieces of music together. Within a piece of music we have music theoretic constructs such as key, tempo and meter, which are defined globally, and time localized structure such as pitch, tone, harmony and rhythm. At this local level, we can begin to represent the structure in terms of physical entities, such as frequencies, transients and noise. A complete representation of musical information can be naturally expressed as a *hierarchical Bayes model* [Gelman, 2004]. For example we could express three levels of musical information thus described as $\theta_1, \theta_2, \theta_3$ progressively incorporating higher level information, and write the prior as

$$p(\theta) \equiv p(\theta_1, \theta_2, \theta_3) = p(\theta_1|\theta_2, \theta_3)\, p(\theta_2|\theta_3)\, p(\theta_3)$$

When we are able to draw random samples from the likelihood and prior distributions we may simulate data y under various conditions and assumptions. Such a scheme is known as *generative* modelling. This allows us to evaluate the modelling assumptions not only using mathematical considerations, but also perceptually by listening to the generated data and assessing it qualitatively. For instance, a generative model for the frequencies contained in a model of musical pitch (which is a perceived characteristic of a musical note) would include definitions of fundamental frequency and harmonicity. As inharmonicity can be substantial in some musical instruments, we would be tempted to put a vague model on how the frequencies of individual harmonics are related to the fundamental. If the model is too vague, we would be able to perceive by listening to simulated data that the perceived pitch is no longer equal to the desired pitch.

1.2 Scope of Work

The signal processing and modelling of musical signals is a diverse field, with a multitude of approaches, philosophies and goals. Here we present a brief overview of popular approaches, and how they relate to the methods we develop in this thesis.

1.2.1 Psychoacoustics and auditory modelling

These approaches characterize and model the human auditory system. Principles of psychoacoustics underlie present audio coding standards [Ahlzen and Song, 2003]. Music is not merely the production of sound by physical processes, but also includes how these sounds are perceived. Hence any system which processes musical signals, even when based solely on physical models, should consider the perception of musical sounds. We might ask whether an automatic music transcription system should transcribe notes which are not audible for instance.

Principles known from psychoacoustics include:

- The frequency and dynamic (intensity of sound) response of the ear, including the range of sounds that can be detected and the resolution with which they are perceived. Physical measurements of different parts of the ear, (from canal to neurones) have been modelled using filter banks and other non-linear signal processing techniques, to compute a summary spectrum, which maps higher order harmonics to lower order harmonics, partially accounting for the ear's ability to perceive pitch. Currently in some Bayesian systems, the periodogram is used as a spectrum estimator to guide Bayesian inference to more likely frequencies and pitches in the signal: a proposal distribution in an MCMC setting, see

Section 4.2.2. It is reasonable to suggest that a spectrum estimator based on an auditory model and pitch perception could improve existing methods.

- Masking: a psychoacoustical effect when tones (for example, musical pitches) are presented in such a way that one tone renders the other(s) inaudible [Gelfand, 2004]. For example, a loud tone can mask a quiet tone with a similar frequency or pitch. A related phenomenon is the fusing of two similar sounds into one, such as two tones with equal loudness and similar frequency.

- Perception of the parameters of musical notes such as pitch, loudness and harmonicity. Pitch and loudness for example are not perceived linearly, or independently one of another. The Bark scale [Zwicker, 1961] is a subjective scale of loudness, and the Mel scale [Stevens et al., 1937] is a perceptual scale of pitch. A well known phenomenon in the perception of harmonicity is the ability of the human ear to reconstruct missing fundamental frequencies [Todd and Loy, 1991] for example when listening to a bassoon.

1.2.2 Machine learning techniques

We may describe an example of this type of approach in general terms as a two-stage procedure:

1. Extracting salient features from frames of audio. The method used to extract the feature depends on the application. Applications which aim to extract frequency related information, such as pitch or harmony, from the music, may start by computing the Fourier transform. A popular feature computed via the DFT is a chroma vector [Wakefield, 1999], which describes the energy distribution between the 12 notes of the Western pitch scale. This information can then be used in a chord recognition [Papadopoulos and Peeters, 2007] or key recognition [Peeters, 2006] applications, by reasoning that notes with higher energy distributed to them will often form a root, third or fifth of the chord. Feature extraction methods which prove successful and useful for a musical signal processing application can often inspire or be adapted in a generative model approach. Considering the example of chroma vectors, the expected distribution of energy across different note groups can be treated as a Bayesian prior for chords and keys in music.

2. Learning and classification algorithms. These algorithms map the features as inputs to the information to be extracted from the music as an output. The majority of these algorithms were not designed specifically for musical signals, but rather map real-valued vectors to labeled classes. The distinction between the feature extraction method and the machine learning algorithm may be considered analogous to the separation of a probabilistic model and the inference algorithm, although machine learning algorithms do assume models of both the features and the labels. Some popular algorithms and their uses include support vector machines for classification [Burges, 1998] and dynamic programming [Rabiner and Juang, 1993] for aligning sequences of music.

1.2.3 Generative models

The scope of a generative model for music, i.e., the actual data being modelled, varies. A model which describes the production of each sample and channel of a digital audio signal may be desirable for high

fidelity signal processing, however sample rates of 44.1 kHz (CD quality) or 48 kHz (digital audio tape) may be unmanageable in terms of computation and inference. Instead, a generative model is usually defined on a simpler domain, e.g., overlapping frames of audio, and additionally some preprocessing and downsampling may be applied using the Fourier transform. These effects however are often non-invertible, for example taking the magnitude of the Fourier spectrum, and losing the phase continuity between frames of audio. For the signal to be reproduced without audible artefacts then requires additional postprocessing such as using a phase vocoder [Flanagan et al., 1965]. The modified discrete cosine transform (MDCT) which has a 50% overlap between frames retains the phase continuity between frames [Princen et al., 1987] hence its inclusion in the MP3 standard.

In this thesis we have chosen to build upon approaches that utilize Bayesian and generative modelling ideas for music. Crucially at the lowest level we restrict ourselves to models proved to be valid for audio signal processing, for in a hierarchical Bayes model musical information is represented by additional levels of hierarchy above that existing for audio signals, because musical signals are a subset of audio signals.

For this reason we have adopted a bottom-up approach to the design for our models, and the structure of this thesis is similar. Most Bayesian analysis is carried out in a similar manner: beginning with the definition of the likelihood, and adding levels of hierarchy until the model is deemed sufficient for the purpose. The models that we propose and develop here are not a complete representation of music information. We have focused here particularly on the modelling of pitch and temporal structure, such that we can infer the positions of musical notes both in time and frequency, and their respective dynamics. The information we obtain may be stored in the intermediate MIDI format for musical events. MIDI (Musical Instrument Digital Interface) is a widely adopted standard available at http://www.midi.org for controlling electronic musical instruments. The data transmitted through a MIDI interface does not contain the recorded waveforms used to synthesize the music into audio, and therefore compactly represents musical content.

1.3 Motivation

The motivation for this work is not directly related to the classification of music and music recommendation systems. The technology behind these systems is advanced and has led to the development of commercial applications, for example www.Last.FM which is an Internet radio, social network and music recommendation service. That technology does not necessarily involve signal processing or machine listening, as there is abundant information supplied by human listeners, for example in the form of *tagging*. Listeners are often very willing to provide such information and even gain utility from it, which contributes to the success of these systems. However there is interest in the automatic generation of tags, see for example Eck et al. [2007].

The classification and identification of structure *within* a piece of music is perceived to be much more laborious, time consuming, subjective and frustrating. Such is the task of a trained musician in music transcription: to write the musical score after listening to extracts of the audio multiple times. Unsurprisingly there are few sources of musical audio labeled with a corresponding accurate MIDI transcription readily available.

Transcription is not necessarily the final goal: being able to align a MIDI file with a performance in the audio is also appropriate for the applications we envisage, as there is a large quantity of MIDI files related

to the score of a piece of music, but however divorced from a real performance of that piece.

Content-based music information retrieval (MIR) is possible using the models presented here. Some examples of how a system may be used include:

- Notation of improvised music, for example in Western Jazz
- Preservation and study of music from traditions without a notation system, which instead rely on oral transmission.
- Detailed performance analysis for musicology [Scheirer, 1998, Seashore, 1936]. One example of this is the Mazurka project[1], which has made detailed annotations of the tempo and dynamics of recordings of Chopin's Mazurkas by several performers. These annotations have been used to compare and contrast different styles and approaches to playing the same piece, capturing performance effects such as phrase arching [Cook] and accentuation patterns. This approach has been used to identify historical recordings which have influenced today's performers, and even cases of copyright violation where existing performances have been replayed.
- Visualization of musical structure in a music media player framework. The Sonic Visualizer [Cannam et al., 2006] allows musicians to listen to and study recordings, offering synchronized playback and display of annotations (including MIDI) and frequency spectra. This tool is already integrated with an audio alignment system called MATCH [Dixon and Widmer, 2005].
- Score alignment - matching events in a score with corresponding audio cues in a recording. The two applications mentioned above would be improved with automated score alignment, reducing much of the manual labour involved.
- Score following - tracking the position of a live performance in a score. One motivation for this application arises in present day music compositions which rely on the synchronization of human performers with prerecorded or synthesized electronic music. The current score follower at IRCAM is trained during rehearsal time to improve its performance, see for example Schwarz et al. [2005]. Current development is centered on anticipatory score following, see Cont [2009], permitting a higher level of interaction between the performance and the score follower. A similar development is the Music Plus One system [Raphael, 2006] which uses score following to accompany and anticipate a performer. The system is used for example to control the playback of a Music Minus One[2] recording to assist musicians when practicing a piece of music without live accompaniment.

As we are using generative models, we have also the following applications:

- Source separation. Score guided source separation may be used to produce the Music Minus One recording automatically from a favoured historical recording [Raphael, 2008], a process known as *de-soloing*. Another use of source separation for musical signals is the separation of polyphonic instruments (for example piano and guitar) into separate channels based on pitch or register, which in a recording studio would be typically recorded on a single channel. This functionality has been demonstrated in Melodyne's Direct Note Access technology[3].

[1] `http://www.mazurka.org.uk`
[2] A recording of a piece of music for soloist and accompaniment where only the accompaniment is recorded
[3] `www.celomony.com`

- Object coding based music compression and synthesis. As discussed in Plumbley et al. [2002] the recent MPEG-4 standard for structured audio [Vercoe et al., 1998] provides for an advanced form of parametric coding of musical signals. The labeling required for the coding could be produced with automated music transcription, and the decomposition and synthesis tasks could also be automated using source separation.

- Morphing, reconstruction and digital effects. Bayesian models have been used for audio reconstruction [Godsill, 1997, Cemgil and Godsill, 2005], noise estimation [Godsill, 2009] and enhancement [Wolfe et al., 2003]. These approaches can be readily modified by extending the prior models for general audio to account for the higher level of structure present in music (for example harmonicity). These extensions form a major part of this thesis.

1.4 Outline of Thesis

This introduction has described the motivation and approach to the research covered in this thesis. We have demonstrated our reasons for adopting a generative modelling approach using Bayesian inference, but have reflected on how complementary approaches from psychoacoustic modelling and machine learning can be adopted. The three following chapters provide the background and foundation for our research.

Chapter 2 provides an overview of our current knowledge of the structure of musical audio signals, drawing from music theory, the physics of musical instruments and the propagation of sounds, and psychoacoustics. The material covered in this chapter is the basis and reasoning for having selected the particular Bayesian priors that we use in this research.

Chapter 3 reviews the literature for *machine listening* of musical signals. Since 2004 a community driven evaluation of machine listening systems for the important applications in this field, known as MIREX)[4] [Downie, 2008], has become prominent. Previously there had been no overall consensus on standardized data sets or evaluation metrics to use for comparing the performance of different systems. In Chapter 3 we use this resource to learn how this area of research has developed and has been influenced over the last few years, and make an objective comparison of the current state-of-the-art technologies.

Chapter 4 introduces the field and methods of Bayesian signal processing, reviewing models and inference methods. Models and inference algorithms used in this thesis will be covered in greater detail in this chapter so that these are collected into one place and are referenced throughout this thesis where they are used.

The remaining chapters describe the new research carried out. Each chapter describes firstly a novel Bayesian model for some level of musical structure in audio. Secondly, a variety of inference methods for these models are developed, and finally the applications of each model, and results presented using data-sets from the literature review in Chapter 3. The chapters are not fully self contained, as the models used are extensions of models already described in the thesis or in the literature, but the focus of each chapter is on one particular level in a hierarchical Bayes model of musical audio, and the chapters are ordered accordingly: partial frequencies and amplitudes in Chapter 5; the grouping of partials into musical notes in Chapter 6; the harmonic and temporal distributions of time-frequency basis coefficients in Chapter 7; prior distributions for the volume of notes and their continuity in Chapter 7; and finally dynamic models for tempo and how the performance of a piece of music moves through the score of that piece in Chapter 8.

[4] `http://www.music-ir.org/MIREX`

Chapter 5 describes a generative model for musical audio using the analytic representation of a signal. This model is based on existing Bayesian models for musical audio using sinusoidal and Gabor bases, but modelling the analytic representation has a number of implications for both inference and the prior structure. We also consider common musical effects such as frequency modulations (vibrato) and amplitude modulations (tremolo) and model these in musical notes as multiple partials occupying the same harmonic position. We then use the model and inference methods developed to perform spectrum estimation and polyphonic pitch transcription in musical signals, demonstrating its ability to model vibrato and detect higher order partials.

Chapter 6 describes a Poisson point process model for inferring musical notes and chords from partial frequency estimates. The model allows for multiple and missed partial detections, and can be applied to the model described in Chapter 5 and also other spectral estimation schemes, both Bayesian and heuristic. The primary advantage of this model is that calculating the likelihood function is computationally inexpensive and inference is straightforward. A simple and intuitive prior is used with a partial estimation scheme using Bayesian model selection on the model in Chapter 5 to produce an effective system for inferring polyphony in short frames of music.

Chapter 7 describes modelling the coefficients of a time-frequency transform of a musical signal by variance parameters. The variances are grouped into a matrix, which is composed of harmonic factors across frequency and excitation factors across time, analogous to non-negative matrix factorization (NMF). A number of Bayesian inference procedures are proposed for rapid and efficient inference, which is necessary for processing large amounts of audio data. A number of applications for general musical signal processing are illustrated using these models. We extend the model with prior structures for the volume of a note, and its onset and offset. This allows the inference of a MIDI transcription from musical audio. The transcription performance of the model is compared to existing systems using a large selection of synthesized classical piano music.

Chapter 8 develops two models for inferring the motion of a hypothetical 'score pointer' through the performance of a piece of music. The first model is a hidden Markov model with a tempo variable describing the probability of moving from one position in the score to the next. The second is a Poisson model counting the expected number of note onset and offsets occurring at each point in the music. Results are presented for score following and query-by-tapping music retrieval applications.

Chapter 2

Structure of Musical Audio

2.1 Introduction

In this chapter we develop some understanding of musical audio which will aid us considerably in defining models. This chapter covers known and experimentally derived results from physics, psychoacoustics and musical theory, which are a necessary foundation for the rest of this thesis. In Chapter 3 we cover the progress made in using these models for the applications grouped under the encompassing term *machine listening*.

In analyzing music audio, we need to address both the physical production of the sound and how it is perceived. Most systems for the analysis of musical audio model either the musical instrument or the auditory system in isolation. In reality neither exists in isolation. Sound is produced by the instrument and received by the sensory system (which includes but is not restricted to the auditory system, as the existence of deaf musicians proves). Feedback may exist in the form of performance, as indicated in Figure 2.1 on page 20.

Physical models of musical instruments are well studied, and although this area of research is by no means dormant or redundant; much of what has already been discovered is of value to us. Fletcher and Rossing [1998] reviews the physics of musical instruments, and forms the basis of our description of pitched musical instruments in Section 2.3.

Models of the perception of sound focus firstly on the auditory periphery and secondly on psychoacoustical studies. We will not describe in detail the research in these areas, pointing the reader to Klapuri [2006] for an excellent introduction; but focus on the models that have been developed as a result of this research.

2.2 Perception of Musical Audio

The process of perceiving audio can be divided into two stages. The first, physical, stage converts the pressure waves that transmit sound through the air into electrical signals in the brain. Central to this is the action of the *basiliar membrane*, which is a stiff structure in the inner ear separating two fluids: the endolymph and the perilymph. One function of the basiliar membrane is frequency dispersion: the membrane varies in thickness and stiffness, thus it responds to different frequencies across its length. The variation of location

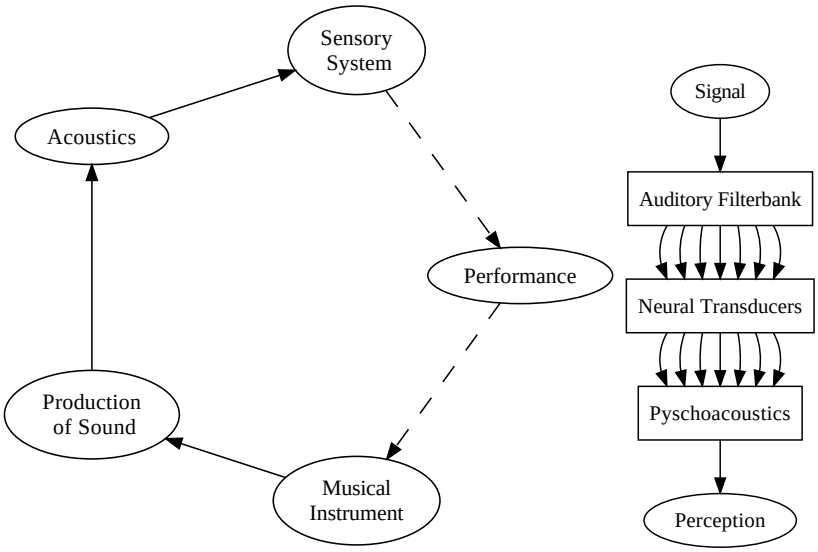


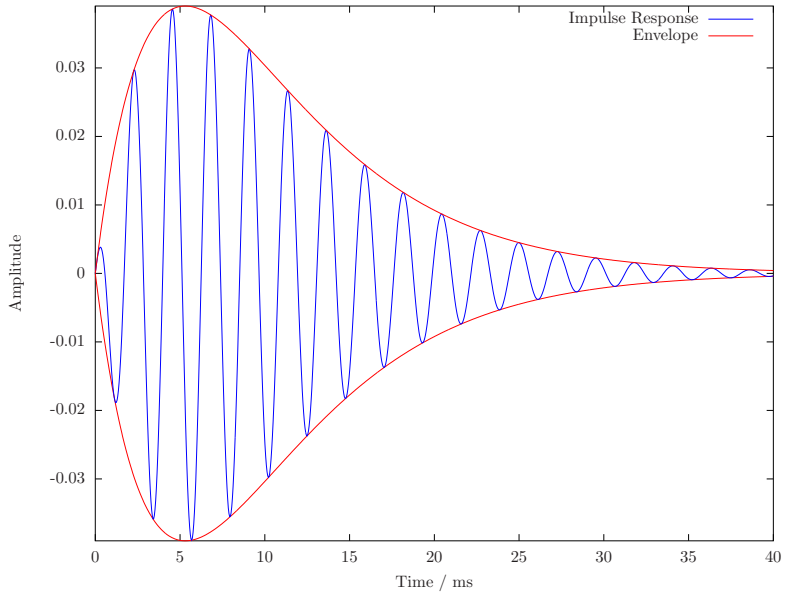
Figure 2.1: The flow of information in musical production and the auditory system. Sound is produced by a musical instrument and received by the sensory system. Performance is a feedback route from the sensory system back to the instrument. The flow of information through the auditory system is shown here as a block diagram. In reality many more channels exist than are shown here.

with frequency is described by the Greenwood function [Greenwood, 1961].

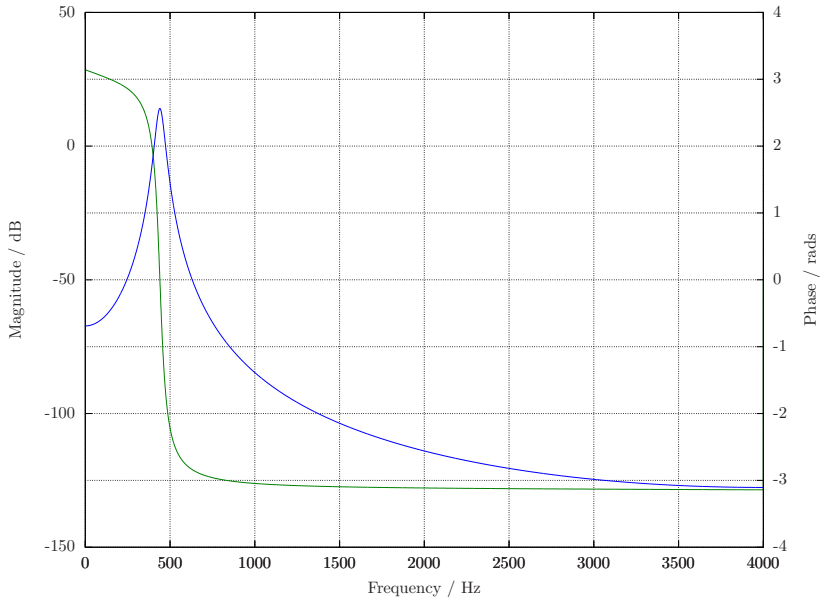
The auditory periphery may be modelled by a filter bank, splitting the received signal into a number of channels (Figure 2.1 on page 20). Each filter has a bandwidth selected to model the frequency selectivity of various parts of the basiliar membrane. Experimental results (Patterson et al. [1992]) suggest that gammatone filters are a good approximation to the frequency response. Figure 2.2 on page 21 shows the impulse and frequency response of a second-order gammatone filter. Each channel is then subject to dynamic level compression, half-wave rectification and low pass filtering. These processes are designed to model neural transduction. The dynamic level compression models the loudness response of the ear which is relevant in our discussion of dynamics in Section 2.2.1. The frequency content is then typically analyzed by computing the *summary spectrum* which is the summation of the spectrum magnitudes across the channel outputs after the low pass filtering operation.

A side effect of the half-wave rectification combined with low pass filtering is that the harmonic complexity of a musical signal is reduced: higher order partials are mapped onto lower order partials (see Section 2.3 for an explanation of these terms) which may explain why humans are good at perceiving multiple pitches. One difficulty with applying standard spectral analysis even for monophonic signals is that the fundamental frequency does not necessarily have the largest amplitude. However when the mapping of partials takes place and we observe the summary spectrum (see Figure 3.1b on page 33 for example), the fundamental does then have the largest amplitude and can be easily identified. A complete auditory model based on the process outlined above appears in Meddis and O' Mard [1997].

The second stage of perception is psychological, occurring within the brain. The first research on the frequency response of the ear was carried out in Fletcher and Munson [1933] where contours of equal subjective loudness (measured in phons) are obtained for different frequencies and sound pressure levels.



(a) Impulse response and amplitude envelope



(b) Magnitude and phase of the frequency response

Figure 2.2: Impulse and frequency response for a second-order gammatone filter with centre frequency 440Hz and sampling rate 8000Hz

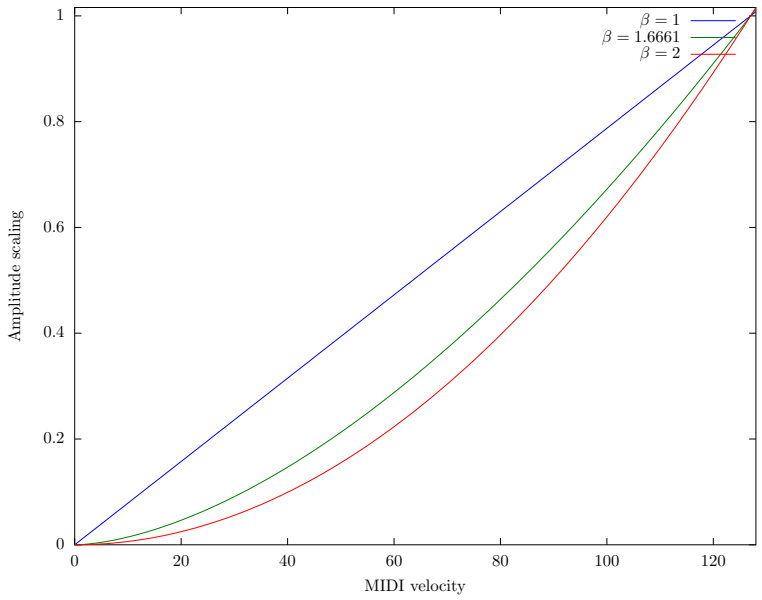


Figure 2.3: Volume curves (2.1) used by some MIDI implementations. $\beta = 1$ gives a linear response, Roland's GS standard uses $\beta = 2$, a square-law relationship, and $\beta \approx 1.661$ is derived from the rule of thumb that loudness doubles when the sound intensity is increased by a factor of ten

2.2.1 Dynamics (Loudness)

Dynamics refers to the volume of a musical note, which may have stylistic interpretation, but primarily refers to the NOTE ON VELOCITY MIDI event. Typically a particular dynamic notated in a musical score e.g., *forte* (loud), is mapped to a range of velocities.

The mapping of velocity to a scaling of the note in amplitude is relevant to underlying signal models. Based on the human ear having a perceptual logarithmic response to amplitude variations, GM (General MIDI) synthesizers use the following volume curve

$$a = \left(\frac{v}{127}\right)^{\beta} \tag{2.1}$$

which expresses the amplitude scaling a in terms of the note velocity v as a fraction of the maximum velocity 127 allowed by the MIDI standard, and a logarithmic response scaling term β. Figure 2.3 on page 22 plots volume curves for the different values of β for implementations of the MIDI standard.

The RWC database [Goto et al., 2003, Goto, 2004] also contains samples at three levels of dynamics: *forte, mezzo, piano*. Studying the musical instrument samples here can give us a granular yet instructive set of volume curves for different instruments.

The dynamic of a note is not necessarily constant across the duration of the note. *crescendo* (gradually getting louder) and *diminuendo* (gradually getting softer) are typically modelled by synthesizers as linear trajectories in note velocity. *tremolo* is a periodic variation of volume, which occurs often with *vibrato* (2.3.4), the pitch analog of volume oscillations.

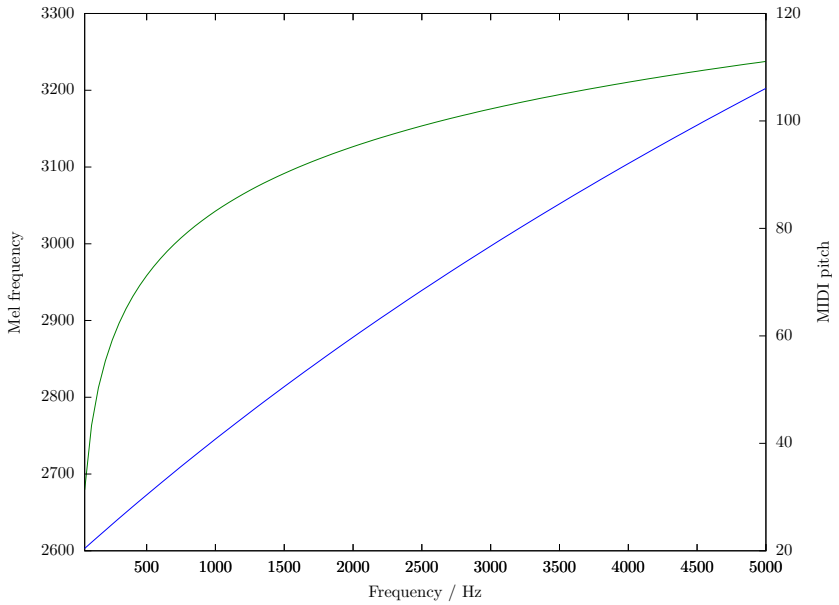


Figure 2.4: Comparison of the Mel frequency scale (2.2) and the MIDI definition of pitch (2.3)

2.2.2 Pitch

Pitch is the perceived fundamental frequency of a musical note. This perception can deviate substantially from physical reality, especially when the sounds are not quasiperiodic. Experimental studies have shown that missing fundamentals and harmonics are tolerated, as is a substantial amount of inharmonicity [Plack et al., 2005]. Hence we typically distinguish between pitch estimation and fundamental frequency estimation. Pitch estimation may not therefore directly rely on physical signal models, but rather on a more approximate representation of a fundamental frequency.

One strong argument in favour of this is that a listener is able to group musical notes by pitch independently of the timbre of that note. A listener is able to identify a piano, a bell, a tympanum and filtered white noise as having a common pitch. Moorer [1977] defines the pitch of a sound as some property that allows it to be matched to a sine wave with a particular frequency. A sine wave itself has no harmonic structure whatsoever, however it has an identifiable pitch. Hence although the harmonic structure of a musical note may be invaluable to identifying the pitch of the note because the instrument is approximately a harmonic oscillator (Section 2.3), harmonicity is not an adequate description of pitch itself.

Pitch perception is not independent of the volume either. The Mel frequency scale [Stevens et al., 1937], which was arrived at by comparing perceived intervals of equal pitch with frequency intervals, illustrates that the perception of pitch is non-linear. A general rule is that that pitch decreases with increased loudness. One formulation of the Mel frequency scale is

$$m = 1127 \log_e (f/700 + 1) \tag{2.2}$$

As with many parametric models in signal processing, pitch estimation may be approached in the time

domain or in the frequency domain. Autocorrelation function (ACF) based methods detect periodicity in a non-linear transformation of the Fourier spectrum of a signal. As such, regularly spaced partial frequencies in the spectrum are detected by an increase in the energy distribution around those frequencies. Comb filters add a delayed copy of a signal to itself, causing periodicity in the signal to be constructive when the delay is correctly specified. The lag at which the maximum of the ACF or the comb filter response occurs corresponds to a dominant fundamental frequency in the signal, and is used as an estimate of the pitch, as illustrated in Figure 2.5 on page 25. Klapuri [2004] shows that comb filter solutions exhibit both better SNR and pitch detection range than ACF methods, but at the cost of additional computation.

Despite our above discussion concerning the imprecise relationship between pitch and fundamental frequency, we still need to mention that the MIDI standard has adopted the following formula to map between Western tonal musical pitches and fundamental frequency:

$$p = 69 + 12 \times \log_2 \left(\frac{f_0}{440} \right) \tag{2.3}$$

where p is a pitch such that there are 12 integer pitches per octave, and $69 = A440Hz$ is the standardized pitch. A *semitone* is the difference between two pitches, and a *cent* is 1/100 of this interval.

2.2.3 Timbre

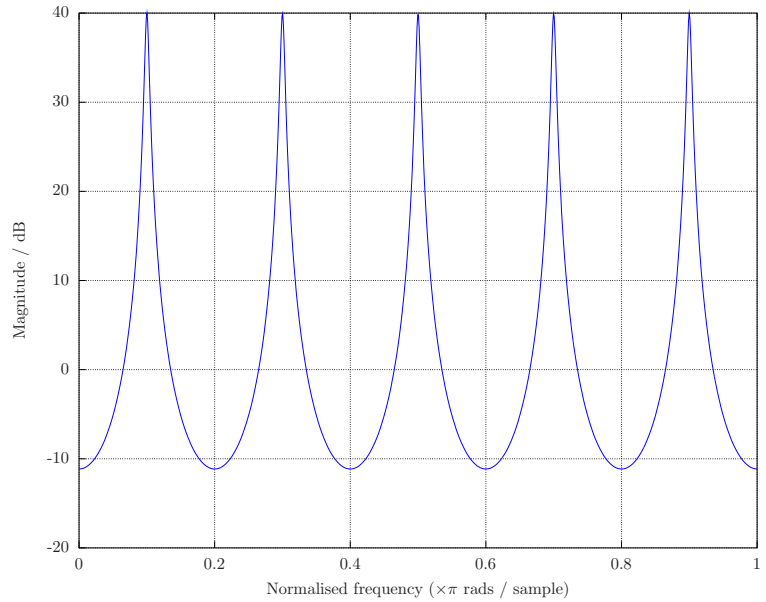
Timbre is another grouping category of music perception which can stand independent of dynamics and pitch. In terms of frequency analysis, because pitch refers to a specific frequency, therefore timbre may only refer to the overall spectral profile. This is backed up by studies which have shown that timbre is related to the relative amplitudes of the partials. Inharmonicity in the partials also affects timbre. Away from frequency domain considerations, the timbre also can be characterized by the onset of the note, which is typically unpitched due to the non-linear activation mechanism to generate the sound.

2.3 Oscillators

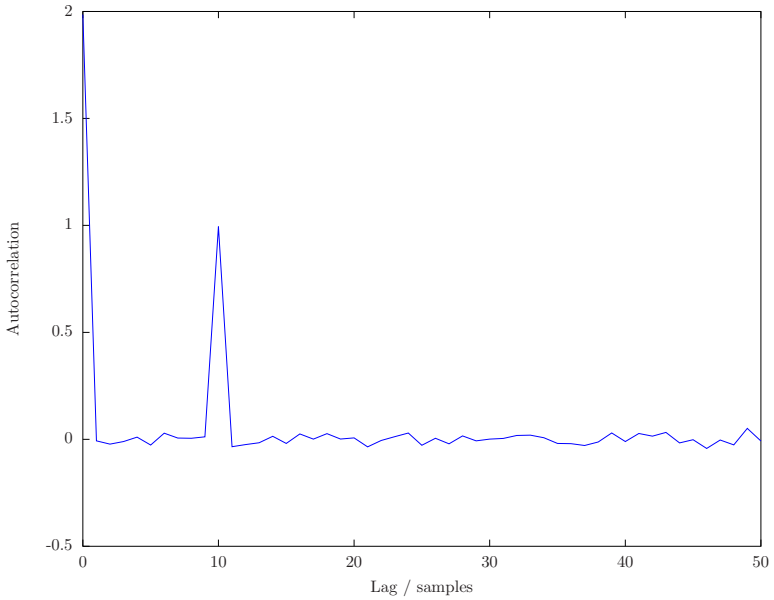
The perception of a pitched sound requires some periodicity in the signal. The most common means of producing periodicity is by a mechanical system known as an oscillator. Oscillators are used in a great variety of musical instruments.

2.3.1 Terminology

The fundamental frequency often corresponds to the lowest partial frequency in a signal, which may be the oscillation across the entire string length for a string instrument, or the air column for a wind instrument. Other partial frequencies occur at the resonant frequencies of the instrument. Because of the relative spacing of the resonant frequencies, partial frequencies are often integer multiples of the fundamental frequency. Inharmonicity is often measured as the deviation in cents of a partial from the closest ideal harmonic of the fundamental frequency.



(a) Magnitude response of the comb filter



(b) Autocorrelation of a white noise signal filtered by the comb filter

Figure 2.5: Frequency response and autocorrelation of a comb filter with 10 samples delay. The maximum of the autocorrelation occurs at zero lag, but the peak at 10 samples corresponds to the fundamental frequency of the signal.

2.3.2 Stringed Instruments

For a string to be a harmonic oscillator the requirement is that it must be homogeneous, infinitely thin and flexible. In this case, a string will exhibit a fundamental frequency related to its length and partial frequencies at exactly integer multiples of the fundamental.

This is rarely the method used to construct a stringed instruments simply because the space required for the string lengths and their transverse vibrations is impractical for low pitches. In these cases, the stiffness of the string is increased, so that the string vibrates more slowly. The result of this however is that inharmonicity is introduced. Stiffness is increased by using thicker strings, or winding strings, or increasing the tension applied to the string. Hence for stringed instruments such as the piano, guitar, harp and members of the violin family played *pizzicato* (plucking the string), inharmonicity is present. For a piano, the following formula relates the harmonic frequencies f_h to the fundamental [Fletcher and Rossing, 1998]

$$f_h = h f_0 \sqrt{1 + h^2 B} \tag{2.4}$$

where B is the inharmonicity coefficient.

In the case of a bowed violin family instrument, the non-linear action of the bow on the string negates the effect of the inharmonicity. The partial frequencies are driven to be multiples of the fundamental frequency. The fundamental however is not precisely equal to the length of the string, a fact which is taken into account when a violinist plays the instrument.

2.3.3 Wind Instruments

The resonator in a wind instrument is an air column, the length of which is controlled by various means. A partially open pipe has end effects where the acoustic length of the pipe differs from the geometric length. However the effect varies with frequency, giving rising to inharmonicity.

There are exceptions: flutes and clarinets have been experimentally determined to have harmonic partial frequencies. This is due to the sound-generating mechanism not being linear whilst the resonator itself is (approximately) so. The system exhibits mode locking behaviour which forces partial frequencies to assume their ideal positions.

Other interesting points about wind instruments are that the fundamental frequency of a bassoon is nearly always absent; and even-numbered harmonics of a clarinet are suppressed because of the cylindrical shape of the resonator [Barthet et al., 2005]. The saxophone, which is related to the clarinet, having a conical resonator, does not suppress the even harmonics.

2.3.4 Vibrato

Vibrato refers to the periodic variation in pitch around a note, and as such is characterized by its *depth*: the amplitude of the pitch variation, and the speed of the variation. Brown and Vaughn [1996] have determined experimentally that the perceived pitch centre of a note with vibrato is the mean value of the perceived frequency of the sound. Vibrato is not purely a frequency modulation, as it is difficult to perform this effect without some degree of amplitude modulation, as shown by Arroabarren et al. [2003]. Deliberate amplitude modulation in the absence of vibrato in a musical performance is known as *tremolo*.

Ratio	Harmonic	Name	Example
2:1	2	Octave	C
3:2	3	Perfect Fifth	G
4:3		Perfect Fourth	F
5:4	5	Major Third	E
6:5	6	Minor Third	E♭
9:8	9	Major Second	D

Table 2.1: Intervals and harmonics

Vibrato is subject to the mechanical limitations of the instrument, physical limitations of the player, and stylistic rules. For some instruments vibrato is not possible to produce. For the violin, Geringer and Allen [2004] have investigated vibrato performance across students, finding that the speed of vibrato was approximately 5.5Hz and independent of the experience of the performer. A more detailed study by MacLeod [2008] suggests that vibrato speed is a function of the pitch and dynamic, whilst the depth is a function of the dynamic only. Similar empirical studies have been carried out for wind instruments and the human voice [Prame, 1994].

2.4 Harmony, Chords and Key

Chords are a grouping of simultaneous pitches, and harmony is the relationship between these pitches to create chords. The dual use of the word "harmony" in describing the relationship between partial frequencies and also notes within a chord is deliberate. Table 2.1 on page 27 is based on the Pythagorean scale of *just* intonation. In practice, Western music has generally adopted a tempered system, where the pitch ratios are approximate, being logarithmically spaced with 12 semitones per octave. The pitch ratios in Table 2.1 on page 27 can be achieved on instruments which do not have fixed tuning.

The seventh harmonic is omitted here as the interval suggested is dissonant (the same is true for the eleventh harmonic). This particular overtone is often avoided in the production of sounds, as it is not sufficiently close to the pitch B♭ related to C.

Consonance is the pleasantness with which different pitches sound together, and it is usually attributed to notes sharing harmonics. One rule for defining consonance is to consider notes in the chord pairwise. If the lowest overlapping harmonic is the eighth or less, the chord is consonant. This accounts for the prevalence of major and minor triads. A chord possessing a semitone interval is always dissonant. The number of possible chords in Western music is reduced massively when semitone intervals are rejected. Overlapping harmonics is thus common in polyphonic music, and is also one of the difficulties for signal processing algorithms to overcome when attempting to resolve separate notes from within a chord.

Favoured chord progressions are those which involve the least number of modifications to pitches present within a chord. For example, chord progressions involving I, IV and V chords are extremely common, each only involving one note out of three to be modified. The concept of a key derives from the chord progressions.

2.5 Tempo

The rhythm of Western music is often felt as a regular occurring pulse, known also as the beat or *tactus*. The rate at which beats occur in music is known as the *tempo*, measured in beats per minute (BPM). Consecutive groups of typically two, three or four beats are known as measures or bars. The properties of this grouping is known as the *meter* of the music. Subdivisions of the tactus, corresponding to the shortest musical note durations (semiquavers for example), are referred to as the *tatum*.

Generative models of tempo are presented in Section 3.4.

2.6 Conclusion

In this chapter we have briefly covered several topics in psychoacoustics and music theory. The goal of this review has been to describe *a priori* structure in musical audio. The technology we review in Chapter 3 and the methods we develop in this thesis all make use of this prior knowledge to infer musical information from audio. In Chapter 3 we will study the applications and implementations of the models presented here for inferring notes and tempo in musical audio.

Some of the results in this chapter may be applied to musical signal processing directly. The auditory model described in Section 2.2 is based on physical evidence and experimental results. When this model is applied in the literature, there are some variations in its implementation, such as the choice of the number of gammatone filters to use in the filter bank, or how the summary spectrum is computed. These variations are motivated by a need for computational efficiency when processing multiple output channels from the filter bank compared to the faithfulness of the model to the ear. On the other hand, only general principles and theories are available from psychological evidence when we consider how the brain perceives pitch. The pitch estimation stage of systems in the literature therefore vary widely in both the model they use and their implementations: for example, compare the use of comb filters and the auto correlation function and the different models they imply. Ultimately we must compare these systems by measuring and evaluating their performance against that of a human listener.

A similar situation, where we have both reliable experimental knowledge and only general principles, is when modelling the sound of a musical instrument. For an inharmonic stringed instrument, we have a model of the harmonic positions of the partial frequencies given by (2.4), with one unknown parameter which can be estimated experimentally [Godsill and Davy, 2005]. However the sound of a note from this instrument may come with a performance effect like vibrato. As described in 2.3.4, experimental studies of vibrato are limited and have only provided us with some limitations on the depth and speed of vibrato, rather than precise measures.

In conclusion we have a set of models describing both the structure of musical audio and how it is perceived and interpreted. These models at present have different levels of certainty because of the process by which they were arrived at. Although we expect models to improve in accuracy with further investigation and experimentation, there will always be uncertainty due to the human element in the performance and reception of music. A flexible musical signal processing system needs to take into account the knowledge covered in this chapter and the certainty we have about it. Moreover, the system should be able to incorporate existing information where it is available. For example, if the key of a piece of music is known beforehand,

for example determined by a human listener, then the system may take advantage of this prior knowledge to estimate chords and chord progressions and consequently improve transcription performance.

Chapter 3

Literature Review

3.1 Introduction

Machine listening of music audio [Scheirer, 2000] refers to the processing of digital audio signals to extract information within a music information retrieval (MIR) context. It differs from, and complements, semantic information which relies on information supplied by human listeners.

As the previous chapter has shown, the modelling of musical audio is not a straightforward task. Physical models of musical instruments are approximate at best, and evaluation using perceptual models is at present still focused on reducing computational expense whilst remaining realistic. Physical models for example may have large numbers of parameters for each instant in time: partial frequencies, amplitudes, noise and so on, making inference using these models expensive until recently. As a result, many authors have resorted to approximations of these models, which have resulted in a wide array of algorithms in the literature.

Moreover, evaluation of machine listening systems is not necessarily straightforward. Evaluation is crucial to performing comparative research and understanding which models and techniques are most appropriate in given situations. Often the presentation of results in publications can be biased by the researcher's particular choice of evaluation criteria, and the conclusions therefore misleading. Taking the example of music transcription, a suitable method of evaluation would involve taking a set of trained musicians and asking them to subjectively assess and rank the transcription outputs of different systems. This type of evaluation involves considerable expense and time, and is not feasible for studies on large libraries of music. An automated approach to evaluation requires choosing libraries of music for which there is a ground truth, such as a reliable MIDI file with its events aligned in time to the audio.

Assuming such a ground truth exists, we need to consider how the performance of a transcription system can be measured. Poliner and Ellis [2007] state that even simple measures such as frame-level transcription accuracy can be biased by reporting too many notes. An event based metric, such as edit distance, or the measure used by the authors mentioned, is more appropriate given that music may be regarded as a stream of note events. The authors also note that, at least in the case of polyphonic piano music which they study, the position of the note onsets is more important than the duration and release of the notes.

Some form of consensus is being arrived at in the form of the annual Music Information Retrieval Eval-

uation eXchange (MIREX)[1][Downie, 2008] contest, which consists of a community-driven open evaluation of research systems on a selection of evaluation tasks deemed to be of interest. MIREX is gaining more credibility in the research community: a good or improved performance at one of the evaluation tasks is now considered acceptable as comparative research.

This chapter will focus on those areas of interest highlighted at MIREX which are relevant to this thesis, and will describe the models and algorithms that have been shown to be suitable to the processing tasks they are designed for. We will demonstrate that the methods which have their basis and justification in the concepts described in Chapter 2 are also the methods which perform best in these comparative evaluations.

3.2 Multipitch Analysis

The goal of automatic music transcription systems is to correctly infer the notes being played in a polyphonic piece of music, producing an intermediate representation, such as MIDI, from an audio track. The inference of polyphonic music may be broken into two tasks:

1. Estimating multiple pitches / fundamental frequencies in individual frames of music. We focus on this task in this section.

2. Tracking the pitches as note contours over consecutive frames, also filtering and smoothing the estimates in the frames. This aspect of multipitch analysis is the focus of Section 3.3.

In MIREX the "multiple fundamental frequency estimation and tracking" task evaluates frequencies as correct when they are within a semitone of the ground truth. The accuracy of the estimation of the fundamental frequencies is not evaluated: a system which reports ideal fundamental frequencies corresponding to the MIDI standard will be ranked equally as one which finely evaluates the fundamental frequencies. For physical models to perform well on such a task requires that the model of the audio fits the observed data very well, and such tractable models have only become feasible in recent years. Auditory models on the other hand will in general perform quite well even with simple implementations, because the definition of pitch is looser than that of fundamental frequency.

3.2.1 Auditory Models

The best performing auditory model, both at MIREX and in other comparisons such as Poliner and Ellis [2007] is that of Klapuri [2008]. The unitary model of pitch perception described in Section 2.2 was first used by Tolonen and Karjalainen [2000] for multipitch analysis using an autocorrelation method. Klapuri [2008] uses a comb filter which analyzes the periodicity of the summary spectrum (see Section 2.2). The output of the comb filter is used to select fundamental frequency candidates. The comb filter weights partial m with fundamental frequency f_0 using the formula:

$$\frac{f_0 + \epsilon_1}{m f_0 + \epsilon_2} \tag{3.1}$$

[1] http://www.music-ir.org/MIREX

where $\epsilon_1 \approx 20$Hz and $\epsilon_2 \approx 320$Hz, and the number of partials is limited to 20. This weighting ascribes more importance to partials with lower harmonic index than higher, as many of these partials have already been mapped to lower frequencies due to the auditory model processing.

In Figure 3.1b on page 33 we compare the summary spectrum which is obtained from the output of the auditory model with the power spectrum obtained using Fourier analysis in Figure 3.1a on page 33. In Figure 3.2b on page 34 and Figure 3.2a on page 34 we compare the salience functions which are obtained from the periodicity analysis by the comb filter. There are pronounced peaks in the salience functions of both spectra at the correct pitches . However the salience function derived from the auditory model correctly ranks the true pitches in first and second position, whereas the power spectrum ranks the third harmonic of one of the notes with a higher salience. This improvement in multiple pitch estimation is due to the aforementioned higher-order partial suppression which is a property of the auditory model.

The estimation of multiple pitches is iterative. At each iteration the pitch with the highest salience is chosen and then the partials corresponding to this frequency are subtracted from the summary spectrum using (3.3). A heuristic formula is used for evaluating the point at which the algorithm should stop, outputting the number of notes that have been detected.

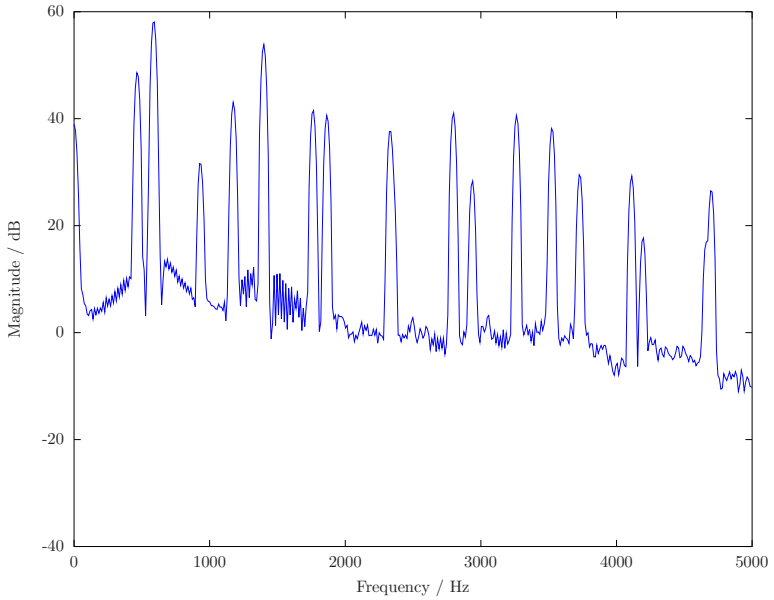
The above model is an example of an iterative estimation algorithm, where a preferred predominant fundamental frequency candidate is determined; and then its contribution to the spectrum is canceled. The algorithm iterates until a stopping criterion is met. This scheme requires less effort than jointly estimating fundamental frequency candidates and also has justification from a psychoacoustical perspective [Bregman, 1990, Hartmann, 1996].Klapuri [2003] refers to this as *predominant F0 estimation*.

3.2.2 Spectrum Estimation

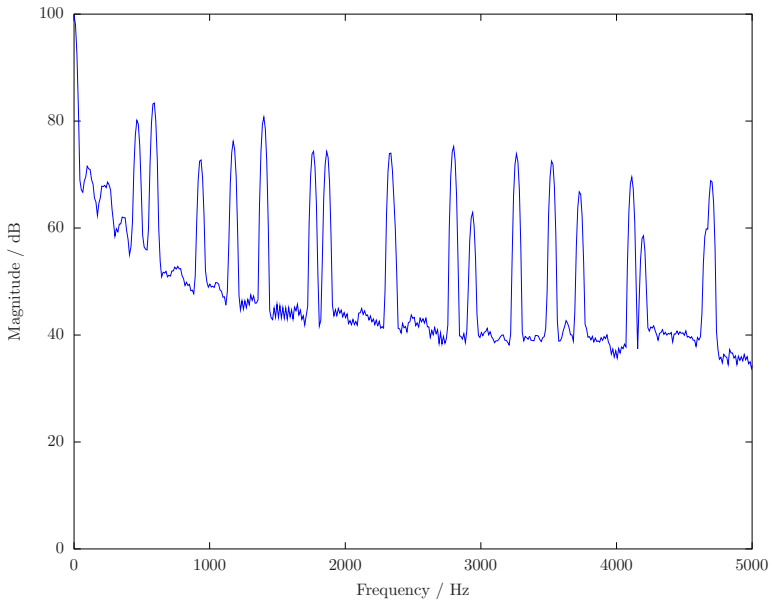
In this section we look at algorithms based on sinusoidal models. In the recent MIREX evaluations of 2008 and 2009 sinusoidal models have shown the best performance, whereas in previous years auditory models outperformed other schemes. Sinusoidal models are used to identify multiple frequencies in musical signals, and the algorithm identifies which of those frequencies are the fundamentals. In early work such as Maher and Beauchamp [1995], estimating multiple frequencies, i.e., spectrum estimation, was performed in isolation to the task of detecting harmonic structure in the frequency estimates and thus identifying the fundamental frequencies. Spectrum estimation has been a popular area of research in the signal processing community for years, and there are many algorithms to choose from. However their ability to detect all of the frequencies of interest in a complex polyphonic signal is limited. The successful approaches that we cover in this section incorporate known priors of musical signals such as harmonicity (3.2.3) and spectral smoothness into their spectrum estimation algorithms.

After obtaining the sinusoidal representation of the signal, the algorithm must then determine which partial frequencies belong to which pitch, and how many pitches are in the frame. The problem of labelling each frequency is combinatorial in nature, especially when the number of pitches is not known, as is normally the case. Hence the algorithm may only perform a limited search through possible combinations of fundamental frequencies.

Spectrum estimation may be carried out in time domain or the frequency domain. The motivation for frequency domain approaches arises from the result by Bretthorst [1989] that the maxima of the periodogram spectrum estimator gives the frequency of a single sinusoid embedded in white noise. Although the result

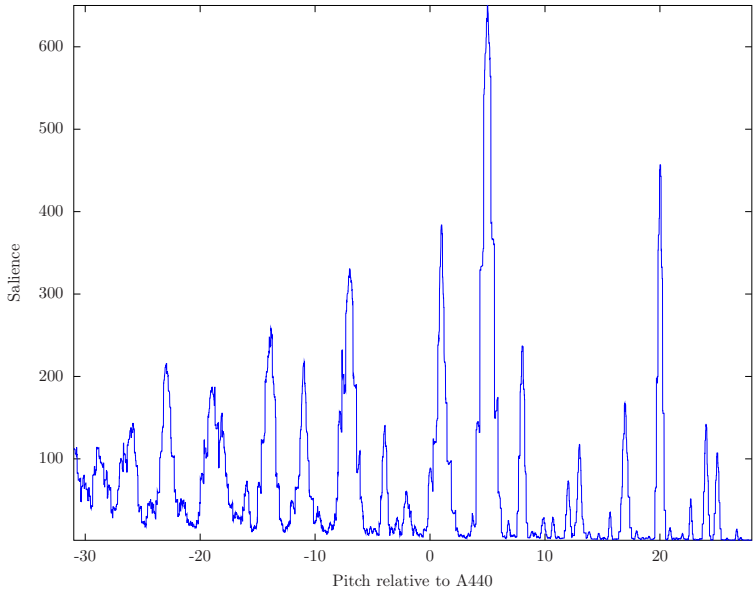


(a) Fourier spectrum for a mixture of two notes with pitch 1 and 5 relative to A440.

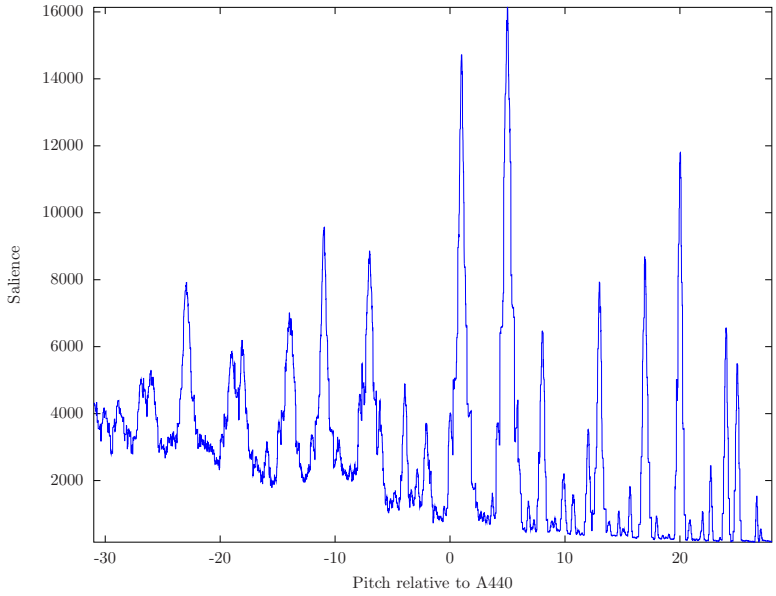


(b) Summary spectrum obtained using an auditory model from the same signal as Figure 3.1a. The higher order partial frequencies have been suppressed due to the half-wave rectification and low pass filtering applied by the model.

Figure 3.1: Comparison of Fourier and summary spectra



(a) Salience function obtained from Fourier spectrum in Figure 3.1a on page 33. Pitch 5 is correctly identified with the highest salience, however pitch 20 (the third harmonic of pitch 5) is ranked with a higher salience than the correct pitch 1.



(b) Salience function obtained from summary spectrum. Pitches 1 and 5 are correctly ranked with the highest salience, even before the contribution of pitch 1 is removed from the spectrum in a iterative-F0 estimation procedure.

Figure 3.2: Comparison of salience functions obtained from spectra

does not hold for multiple sinusoids, a popular approach picks local maxima of the periodogram above a noise floor threshold. One approach for estimating the noise floor in this context is to compute the statistics of the local minima of the periodogram [Martin, 2001], as the minima are expected to be noise components of the signal rather than the sinusoidal components. Bayesian spectrum estimation on the other hand makes use of an explicit signal model, i.e., sum of sinusoids, and a noise model. The principles used to design the systems are similar:

1. Perform spectrum estimation: i.e., identify a number of frequencies present in the noise-corrupted signal. This may be carried out using an explicit signal model, or by extraction from the coefficients in a transform domain of the signal.

2. Identify and label fundamental frequencies and corresponding partial frequencies. A set of possible candidates is generated in this step.

3. Select the candidate set with the simplest explanation.

Stages 2 and 3 may be combined into a single step as the principles used to identify the frequencies are typically rules based on harmonicity and correlation between the amplitudes of the detected partials. For example, Yeh et al. [2005] assumes a sinusoidal model with multiplicative detuning parameters for each partial frequency. The preference expressed by the authors is to prefer candidates with smaller detuning parameters. In a Bayesian setting, this would correspond to a prior which assigned greater probability mass to smaller detuning parameters. Pertusa and Inesta [2008] apply the principle of Gaussian smoothness to the partial amplitudes, which is similar to treating the evolution of the amplitudes as a linear dynamical system.

The preference for the simplest explanation for an observation is encapsulated by Occam's razor, and is naturally applied by means of Bayesian model section (Section 4.1). This principle can be used to guide the design of the models for musical signals, and may be used to select between competing models in a mathematically rigorous manner. It is encouraging to see that signal models are indeed able to outperform auditory models for multiple pitch recognition. One goal of this thesis will be to set the signal models described above in a fully Bayesian framework, using the principles described in Chapter 2. In so doing, we rely heavily on the work carried out for Bayesian spectrum estimation in Andrieu and Doucet [1999] and its extension to polyphonic music in Davy et al. [2006].

3.2.3 Harmonicity

An explicit model for the inharmonicity of stringed instruments is given by (2.4). The inharmonicity parameter is estimated as part of a Bayesian modelling scheme by Godsill and Davy [2005], and an estimation scheme is given as part of the multi-pitch extraction system of Klapuri [2003].

In addition, there exist models for inharmonicity in generic musical instruments, which we review in this section. These models implicitly prefer a lower amount of inharmonicity, that is, partial frequencies are expected to lie close to integer multiples of the fundamental frequency. Another property of the models is that higher frequency partials are allowed to deviate more from their ideal harmonic positions than lower frequency partials, as in (2.4).

Yeh et al. [2005]describe the degree of deviation d of an observed partial f from its expected harmonic position hf_0 as

$$d = \begin{cases} \frac{|f - hf_0|}{\alpha h f_0} & \text{if } |f - hf_0| < \alpha h f_0 \\ 1 & \text{otherwise} \end{cases} \tag{3.2}$$

where α is a tolerance on the amount of inharmonicity allowed.

Godsill and Davy [2005] introduce a generative model for inharmonicity via *detuning* parameters. The detuning of a single partial δ is defined as the multiplicative error term that maps the expected harmonic hf_0 to the observed partial f, i.e.,

$$f = (1 + \delta) hf_0 \tag{3.3}$$

Each δ in the model has a zero-mean Gaussian prior with a constant variance $\sigma_\delta^2 = 3 \times 10^{-8}$. Thus smaller absolute values of the detuning parameters have a higher probability.

3.2.4 Spectral Envelope

In this section we cover how the spectral envelope of musical signals has been modelled in the literature. The relative partial energies of the harmonic series of a musical instrument has an important contribution to its timbre. The partial energies of a musical instrument decay relatively smoothly with increasing frequency, so that the majority of the signal energy is concentrated within the lower harmonics.

A simple Bayesian model for the decay of the amplitudes with increasing frequency is given in Godsill and Davy [2005]. The prior for the amplitude b_m of the mth partial is a zero mean Gaussian:

$$\begin{aligned} p(b_m) &= \mathcal{N}\left(0, \sigma_n^2 \xi k_m\right) \\ k_m &= \frac{1}{1 + (Tm)^\nu} \end{aligned}$$

σ_n^2 is the variance of the ambient noise, and ξ represents the overall signal-to-noise ratio (see Section 5.3 for details of this Bayesian model). The scaling k_m controls the decay of the amplitudes according to a low-pass filter with cut-off frequency T and decay constant ν. The use of low pass filters for modelling instrument timbre is common, for example the system of Karjalainen and Laine [1991]. A fractional delay filter provides the periodicity defining the fundamental frequency of the note, and a low pass filter in the feedback loop causes higher harmonics to decay more rapidly. When excited with a short burst of white noise, the output of the system has a realistic plucked string sound.

3.2.5 Transform Decomposition Methods

The last class of methods we will consider for multi-pitch analysis are those which decompose a linear basis transform, such as the Fourier transform, of a musical signal into separate harmonic contributions from each pitch. The method was pioneered for polyphonic music transcription by Smaragdis and Brown [2003]. In their method, the spectrogram is formed into a matrix $\mathbf{X}$ with the Fourier transform of each overlapping frame of music forming the columns of this matrix, i.e.,

$$\mathbf{X}_{\omega,t} = \left|\text{STFT}\left(t, \omega\right)\right|^2$$

The matrix $\mathbf{X}$ is then decomposed by non-negative matrix factorization [Lee and Seung, 2000] into two factors $\mathbf{X} \approx \mathbf{WH}$ where the number of columns of $\mathbf{W}$ and the number of rows of $\mathbf{H}$ are both equal to the number of pitches that are modelled in this segment of music. Each column of $\mathbf{W}$ models the harmonic profile of a note with a certain pitch. $\mathbf{W}$ may be trained or constrained to some parametric form. Each element of $\mathbf{H}$ gives the weight, or the energy, of that note in a frame. $\mathbf{H}$, when the rows are arranged in ascending pitch order, has the appearance of a piano roll (see Figure 7.4 on page 119 as an example), and may be used as a starting point for a full transcription.

3.3 Polyphony Tracking

The task of tracking multiple pitches present in a piece of musical audio over time is normally evaluated separately from the task of estimating multiple pitches in a single frame (or otherwise stationary section of audio in terms of pitch). This is useful because it is possible to combine different pitch analysis and tracking systems together, and evaluate the performance of each separately.

Two related applications to tracking polyphonic signals are

1. Melody extraction. This refers to extracting a melodic line from a polyphonic piece. What constitutes a melodic line must be answered using music theory, but a practical working definition is that the melodic line corresponds to the predominant pitch (3.2.1) in each frame. We will therefore view melody extraction as a sub-problem within the general polyphonic tracking problem, and do not directly address it here. However, we note that many of the melody extraction systems that perform well in the MIREX task use similar models and algorithms to systems which carry out full multiple-pitch estimation and tracking, rather than being specifically designed for the purpose of melodic extraction.

2. Score alignment. This refers to tracking polyphony through audio when the ordering of the pitches present is known to some extent, but the tempo is not. Hence polyphonic tracking systems should have one model to describe the evolution of the pitches, and another model to describe the evolution of the tempo. We cover tempo models in Section 3.4 as a separate application. In this section we discuss models that describe how the pitches within a polyphonic piece change over time.

Polyphony tracking may be carried out in a real-time online manner, or offline. Offline approaches tend to outperform online approaches, as expected and reported in Robertson and Plumbley [2009], who develop a real-time system based on Puckette et al. [1998].

3.3.1 Attack-Sustain-Decay-Release

An Attack-Sustain-Decay-Release (ASDR) envelope is a function of time present in modern synthesizers used to modulate the loudness of the note. Dodge and Jerse [1997] state that the shape of the ASDR envelope is the dominant factor in the perception of instrument timbre. Thus this relatively simple model is used frequently in the literature for an isolated musical note spanning multiple frames of audio data.

Orio and Déchelle [2001] model each note event in a score using a three state hidden Markov model *attack-sustain-rest*. The duration of each note is governed by the transition probability from the note being in the sustain state in one frame to the note remaining in the sustain phase in the succeeding frame. The

duration of each note thus follows a negative binomial law. A similar technique is used by Devaney et al. [2009] where transient and sustained sections of sung notes are modelled using a three state hidden Markov model.

Cemgil et al. [2006] use a state space model for the evolution of sinusoids through time. The beginning of each note is modelled by the state being drawn from a zero mean Gaussian with covariance matrix modelling the distribution of energy across partials. The sustain/decay phase is treated by each harmonic sinusoid h having a damping ratio $\rho_h = \rho_d^h$ where ρ_d is the damping ratio of the fundamental. In the rest phase, the damping ratio is increased to ρ_r, so that the note rapidly becomes inaudible .

3.3.2 Pitch Evolution

A popular method for polyphonic tracking in the literature is a hierarchical hidden Markov model (HMM). Individual notes are modelled as in 3.3.1. The transition probabilities between notes of different pitches are estimated from large databases of music, such as the work carried out by Ryynänen and Klapuri [2004]. The key of the music is provided as prior information to improve the relevance of the model, and most databases can be shifted cyclically so that data for one key can be applied to another key to improve the diversity of training data. Key detection systems are global, and may rely on coarse feature vectors such as chroma.

3.4 Tempo Tracking

Listeners without musical training are capable of tapping a rhythm corresponding to the pulse of the music, and are able to adapt to changing tempos. With some training, listeners are able to infer the meter (Section 2.5). One goal of machine listening therefore is to be able to infer the same basic rhythmic structure. This task is collectively known as beat tracking, although the emphasis may be more on tracking tempo rather than precise beat locations.

Score following is a closely related application to beat tracking. In score following, knowledge of the score automatically gives us the deterministic relationship between the tatum, beat and metrical structures. Inferring the tatum is adequate for this task. Other than this, the onset detection model of a beat tracker may need to be modified, at least to allow for misses and false alarms. It is also typical in the literature to extend the onset detection model so that it incorporates knowledge of the underlying score, particularly note pitches and volumes. It is an open question whether such extensions generally improve the quality of score following or whether there might be a degradation in robustness to variations in timbre etc. The seemingly satisfactory performance of query by tapping systems (QBT) [Jang et al., 2001] for querying musical databases adds some weight to this question. Moreover, a perceptual evaluation of the alignment of the audio to the score may be based on the alignment of onsets in the music.

In the literature, beat tracking is often coupled with audio onset detection to form a complete listening system. However it is instructive to study these separately, as the onsets may already be available to us in some electronic format, for example MIDI events, and also we may wish to evaluate the performance of different models in a modular system. We will also indicate how these models may be modified as part of a score following system.

Cemgil and Kappen [2003] and Raphael [2001] model the observed onset times y_k as actual onsets τ_k with some added noise ϵ_k per onset. The difference between successive actual onsets τ_{k-1} and τ_k is given

by the expected inter-onset interval (IOI) γ_k (as a multiple of the tatum) scaled by the current value of the tempo Δ_{k-1}, with some additional process noise σ_τ. The tempo evolves as a random process $\Delta_{k-1}, \Delta_k, \ldots$ with process noise σ_Δ. The model can be written in state space form:

$$\begin{aligned} \begin{bmatrix} \tau_k \\ \Delta_k \end{bmatrix} &= \begin{bmatrix} 1 & \gamma_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_{k-1} \\ \Delta_{k-1} \end{bmatrix} + \begin{bmatrix} \sigma_\tau \\ \sigma_\Delta \end{bmatrix} \\ y_k &= \tau_k + \epsilon_k \end{aligned} \tag{3.4}$$

These models solely relate the observed IOIs to the expected IOIs. Beat and metrical information is supplied *a priori* as $p(\gamma_k)$. Figure 3.3a on page 40 presents the model as a Bayesian network .

This model is appropriate for rhythm-based quantization of MIDI events, however we must generalize the model for some detection probability $p(y_k|\tau_k)$ when using audio onset detection. For score following we could extend this model to have a probability distribution over the observed audio $p\left(s_{y_k:y_{k+1}}|\gamma_k\right)$ between the recorded onset times. Raphael [2004] uses a simple generative model for the spectrum of each frame given the score γ_k.

Klapuri et al. [2006] assume a frame-based onset detector, and include metrical structure in their generative Markov model

$$\begin{aligned} p\left(s_k, \tau^{\text{tatum}}_{k-1:k}, \tau^{\text{beat}}_{k-1:k}, \tau^{\text{measure}}_{k-1:k}\right) &= p\left(s_k|\tau^{\text{tatum}}_k, \tau^{\text{beat}}_k, \tau^{\text{measure}}_k\right) \\ &\times\ p\left(\tau^{\text{tatum}}_k|\tau^{\text{beat}}_k, \tau^{\text{tatum}}_{k-1}\right) \\ &\times\ p\left(\tau^{\text{measure}}_k|\tau^{\text{beat}}_k, \tau^{\text{measure}}_{k-1}\right) \\ &\times\ p\left(\tau^{\text{beat}}_k|\tau^{\text{beat}}_{k-1}\right) \end{aligned} \tag{3.5}$$

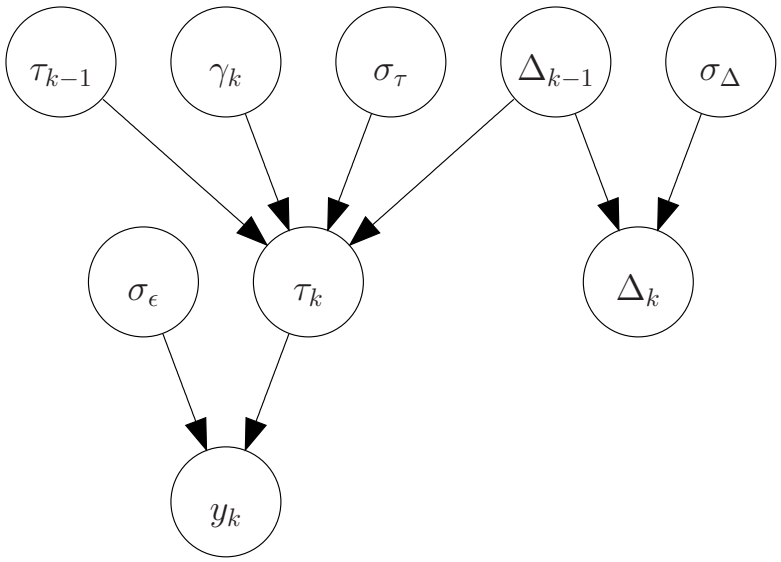
As can be seen from the structure of this model, the fundamental state is the beat or pulse. The metrical structure is derived from the evolution of the beat state. The onset detector itself assigns a likelihood $p\left(s_k|\tau^{\text{tatum}}_k, \tau^{\text{beat}}_k, \tau^{\text{measure}}_k\right)$ for a feature vector s_k based on the frame of audio. Figure 3.3b on page 40 presents the model as a Bayesian network.

Whiteley et al. [2006] adopt a different definition of tempo as the velocity n_k of the tatum m_k over feature vectors s_k of frames of music.

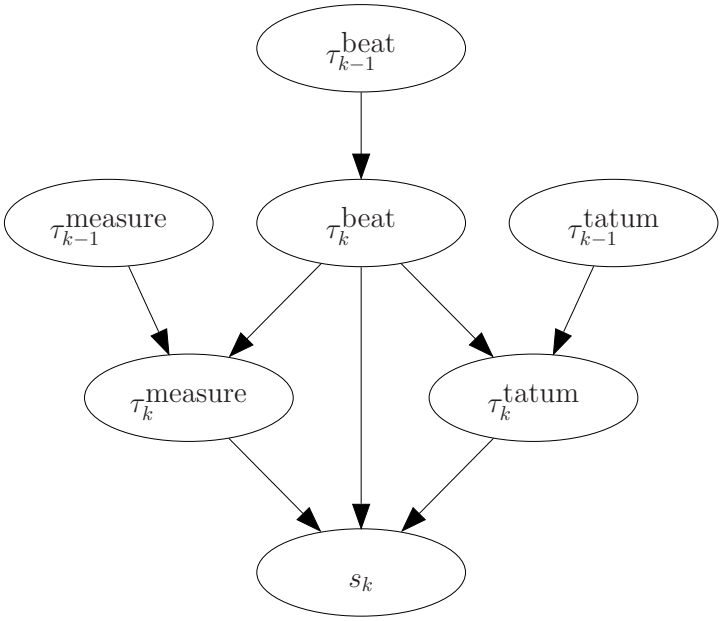
$$\begin{aligned} p\left(s_k, m_{k+1}, n_{k+1}, \theta_{k+1}|m_k, n_k, \theta_k\right) &= p\left(s_k|m_k, \theta_k\right) \\ &\times\ p\left(\theta_{k+1}|\theta_k, m_{k+1}, m_k\right) \\ &\times\ p\left(m_{k+1}|m_k, n_k, \theta_k\right) \\ &\times\ p\left(n_{k+1}|n_k\right) \end{aligned}$$

The $\theta_k = \{M_k, r_k\}$ denotes changes in meter M_k (the number of tatum positions in the bar) and also rhythmical pattern indicators r_k within a bar, hence its inclusion in the onset detector $p(s_k|m_k, \theta_k) = p(s_k|m_k, r_k)$. The tatum moves deterministically according to

$$m_{k+1} = (m_k + n_k - 1) \bmod M_k + 1$$



(a) State-space model of tempo Δ_k with actual onsets τ_k, observed onsets y_k and expected inter-onset intervals γ_k (3.4) for a single time slice k



(b) Markov model of metrical structure (3.5) for a single time slice k

Figure 3.3: Bayesian network representations of tempo models

Both of the above models can be extended to score following by adding a probability distribution of the form

$$p\left(s_{m_k:m_{k+1}}\right)$$

on the frames of audio between tatum positions. Peeling et al. [2007a] extend the model of Klapuri et al. [2006] to score following using a generative model for the spectrogram coefficients.

3.5 Conclusion

Music is inherently hierarchical in structure, and in this chapter we have seen that the state-of-the-art systems for extracting pitches, notes and tempo from musical signals reflect this hierarchy. To extract multiple pitches from a signal, individual partial frequencies are estimated and grouped into the smallest set of pitches that explain the harmonicity and timbre of the signal. Pitch estimates in consecutive frames of music are then linked together as notes. Missing pitch estimates between a note's onset and offset are filled in, and spurious pitches are discarded as noise. Notes in polyphonic music tend to arrive in groups with regular spaced intervals, giving rise to the perception of beat and tempo in music. Concurrently sounding notes also tend to have extended harmonic relationships, giving rise to chords, chord sequences and the key of the music.

It is clear from the above description that a single pass through a musical signal, first extracting pitches, then smoothing pitch tracks to obtain notes, and estimating metrical structure and key, would be incomplete. For example, when spurious pitch detections are discarded as noise, this gives us more information about the structure of the noise in that frame, and can therefore be used to improve the original system that extracted multiple pitches from the frame. An iterative approach thus seems appropriate, however computational performance can be an issue here, particularly if the pitch extraction algorithm only works on a single frame at a time, and frames are then processed in order. Therefore single-pass systems are often experimentally trained offline to determine the optimum set of parameters for pitch extraction and smoothing notes. However today's music has a large variety of genres, instruments, styles and so forth. Selecting an appropriate training set which will generalize well is difficult.

The motivation for an iterative approach has given rise to algorithms based on matrix factorization of time-frequency coefficients, which in single steps can process multiple frames of music, and are computationally viable alternatives (3.2.5). These methods have yet to be shown to be similar in performance to multiple pitch extraction schemes based on auditory or sinusoidal models. Reasons for this include that the models are not physically realizable, as they rely on the spectrogram being additive, the difficulty of determining the number of pitches in each frame, and linking the frames together to track polyphony. These three reasons are the motivation for our research in Chapter 7 and Chapter 8, where we attempt to address each issue without sacrificing computation.

In this chapter we have also seen models for various music phenomena, such as harmonicity, timbre and the envelope of the note energy. Some of these models were derived from studying the physics of the musical instruments that produce the sound, whilst others were developed to capture more general characteristics, such as low inharmonicity and the spectral smoothness of partial amplitudes. The application of these models has been shown in the MIREX evaluations to be effective at extracting multiple pitches from frames of music, and may be cast in a Bayesian setting. In Chapter 5 and Chapter 6 we will consider additional

parameters for frequency and amplitude modulations in a note, and investigate efficient schemes to infer with models containing large numbers of parameters.

Combining a generative multiple pitch model in a frame with a hierarchical hidden Markov model for tracking polyphony and tempo is attractive for musical signal analysis, as the entire model is suitable for Bayesian inference, prior information can be incorporated in a transparent way, and the model is useful for a number of applications. In Chapter 8 we consider models for tracking the movement of a score pointer in a performance, and apply the model to the applications of score following and query by tapping.

Chapter 4

Bayesian Methods for Signal Processing

In this chapter we describe basic Bayesian methods for modelling and inference on which the models developed in this thesis rely on. In Section 4.1 we discuss modelling concepts using Bayes rule, and introduce some popular models and representations of models. In Section 4.2 we describe the inference algorithms applied to these models that are used elsewhere in this thesis.

4.1 Bayesian Modelling Methods

In this section we introduce some probabilistic modelling techniques. 4.1.1 provides definitions and concepts relating to Bayes' rule and comparing probabilistic models as explanations of observed data.. In Section 4.1.2 we show how graph structures can represent complex probabilistic models. We then define the hidden Markov model in Section 4.1.3 which can be used to model causal and dynamical systems. Finally we discuss the exponential family of probability distributions in Section 4.1.4 which are often used in Bayesian approaches because of their useful properties.

4.1.1 Bayes Rule and Model Comparison

4.1.1.1 Parameter Estimation using Bayes Rule

In frequentist statistics, we have observations y produced by a set of unknown model parameters θ according to a likelihood function $p(y|\theta)$. One estimate of the model parameters given the data is the *maximum likelihood* (ML) estimate

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} p(y|\theta)$$

However we often have some prior knowledge expressed as a Bayesian belief $p(\theta)$ concerning the parameters, and $p(y|\theta)$ is interpreted as the information provided by observations y conditioned on the parameters. Bayes' rule states that

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{4.1}$$
$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

which can be viewed as weighting our prior belief with the likelihood of the data to give a posterior estimate of the parameters. The prior captures our belief about the model parameters before we observed any data. The ML estimate is now replaced with the *maximum a posteriori* (MAP) estimator

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\theta|y) \tag{4.2}$$

4.1.1.2 Marginal Likelihood for Model Comparison

The *evidence* or *marginal likelihood* $p(y)$is the normalizing term in (4.1). The term *marginal likelihood* arises as $p(y)$ is the likelihood of the observations y after marginalizing the model parameters θ:

$$p(y) = \int_{\theta \in \Theta} p(y|\theta)\, p(\theta)\ \mathrm{d}\theta \tag{4.3}$$

The marginal likelihood is important for Bayesian model comparison as the only remaining unknown is the identity of the probabilistic model for y itself. A higher marginal likelihood for a model indicates that the model is a better explanation of the observed data. Calculating the marginal likelihood is an application of Occam's razor, which states that a simpler explanation for an observation is to be preferred. Here, simpler implies a model with less parameters. Both ML and MAP estimators will prefer a model with more parameters, a problem known as over-fitting.
We can select the best model for y by computing the marginal likelihood $p(y)$ for every model in the set of models we are comparing. We will illustrate this here with two probabilistic models M_1 and M_2 which are considered possible explanations for y. The two models may have different numbers of parameters. We compute the marginal likelihood for each model, which we denote as $p(y|M_1)$ and $p(y|M_2)$. One method of comparing the models is the Bayes factor [Kass and Raftery, 1995]

$$\frac{p(y|M_1)}{p(y|M_2)}$$

which assesses the evidence M_1 against M_2. A Bayes factor greater than 1 indicates that M_1 should be preferred.

In full Bayesian inference we do not compute point estimates of the model parameters θ such as the MAP estimate. Rather we are interested in inferring the posterior probability distribution $p(\theta|y)$, for which the MAP estimate is the mode of this distribution. Inferring the posterior distribution allows us to compute expectations such as the variance, which gives us some idea of the uncertainty in our parameter estimates, and is important for making decisions based on such inference. Note that as the likelihood $p(y|\theta)$ and the prior $p(\theta)$ are known, then computing the marginal likelihood $p(y)$ is equivalent to computing the normalization constant of the posterior distribution and thus computing the distribution itself (4.1).

4.1.1.3 Generative and Discriminative Models

A generative model for y is a probabilistic model $p(y, \theta) = p(y|\theta)\, p(\theta)$ for which we can randomly sample a set of parameters $\theta' \sim p(\theta)$ and then sample or *generate* an observation $y' \sim p(y|\theta')$. When choosing a generative model to study a signal, we desire that

- Statistical properties of the observed signal should match the properties of data generated by the model
- The parameters of the model θ include information which is to be extracted from the signal
- The prior $p(\theta)$ and the likelihood $p(y|\theta)$ of the model are based on the known physical processes which drive the signal

The alternative to a generative model is a discriminative model which directly defines a probability distribution $p(\theta|y)$ which can then be used to obtain the MAP estimate (4.2). This often means that the original likelihood and prior in (4.1) may not be available in closed form and cannot be used as a generative model. A discriminative model may be designed to give accurate results for the task it is designed for, such as multiple pitch detection, but needs to go through cross validation to ensure that the model will generalize well to unseen signals. For generative models we are able to use model comparison to select the most appropriate model, and model selection can be carried out not only offline on training data but also at the time when the signal is observed.

4.1.1.4 Hierarchical Models

A hierarchical model enforces conditional independencies between the model parameters. There is often good justification in a signal processing application for making this assumption. For example, in a beat tracking application, where we want to jointly track the tempo and the position of beats in a drum track, the beats themselves appear as sudden bursts of energy in the signal, but the tempo is not directly relevant to the observed signal. Rather, the tempo controls the rate at which the beats occur. If we define the parameters related to the beats as θ_b and the tempo parameters as θ_t, then the model

$$p(y, \theta_b, \theta_d) = p(y|\theta_b)\, p(\theta_b|\theta_t)\, p(\theta_t)$$

is not only justifiable from a theoretical point of view, but also can be a generative model: first sample the tempo according to the prior $p(\theta_t)$ and then sample the beats, and finally the signal.

Hierarchical generative models are often represented graphically as Bayesian networks (Section 4.1.2). The graph can then be used to determine how to sample the parameters in turn to generate data from the model, and also guides inference algorithms to estimate unknown parameters given observed data.

4.1.1.5 Conjugate Priors

In this section we define a useful property of certain families of probability distributions which simplifies inference and computation. A family of probability distributions is a set of distributions which share the same functional form but have different parameters. For example, the normal distribution may be parametrized

by mean μ and standard deviation σ

$$\mathcal{N}(y;\mu,\sigma) \equiv \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\mu)^2}{\sigma^2}\right)$$

The mean μ is a location parameter for the normal distribution. A family of distributions with a location parameter has the following functional form

$$f_\mu(y) = f(y-\mu)$$

The standard deviation is a scale parameter for the normal distribution. A family of distributions with a scale parameter has the following functional form

$$f_\sigma(y) = f(y/\sigma)/\sigma$$

Now assume the mean μ of some observed data y is unknown, but the standard deviation σ is known. The posterior of μ is

$$p(\mu|y,\sigma) \propto p(y|\mu,\sigma)\,p(\mu)$$

If the prior $p(\mu)$ is also chosen to be a normal distribution, then it can be shown that the posterior $p(\mu|y,\sigma)$ is also a normal distribution, where the mean and standard deviation of the posterior are given by standard rules (Section A.1). The normal distribution prior $p(\mu)$ is a *conjugate prior* to the mean parameter of a the normal distribution, because the posterior is in the same family of probability distributions as the prior. A conjugate prior is a choice of prior for a particular parameter of a likelihood function, where the posterior of this parameter is in the same family as the prior.

4.1.2 Bayesian networks

Graphical models are a method of visualizing the structure of probabilistic models by diagrammatically representing probability distributions. Inference algorithms can be viewed and defined as messages being passed between nodes of the graphical model [Bishop, 2006]. Directed acyclic graphs, also known as Bayesian networks, are useful for constructing models via conditional probability distributions. Figure 3.3 on page 40 provides two examples of Bayesian networks. Circular nodes represent the values of random variables, such as the unknown parameters and observed data, and edges represent statistical dependencies between the random variables.

Bayesian networks intuitively represent causality. The definition is that an edge is directed from A to B if B is conditionally dependent on A. We call A a parent of B, and B a child of A. The acyclic property makes these graphs suitable for working with *generative* probabilistic models, where we can generate synthetic data by sampling the distributions of nodes with no parents (hence they are conditionally independent of all other random variables in the network), and then moving through the network along the directed edges, sampling each child node dependent on its parents. This is known as *ancestral sampling* [Bishop, 2006]. The probability distribution, and the method of generating samples from it, are represented by the following factorization:

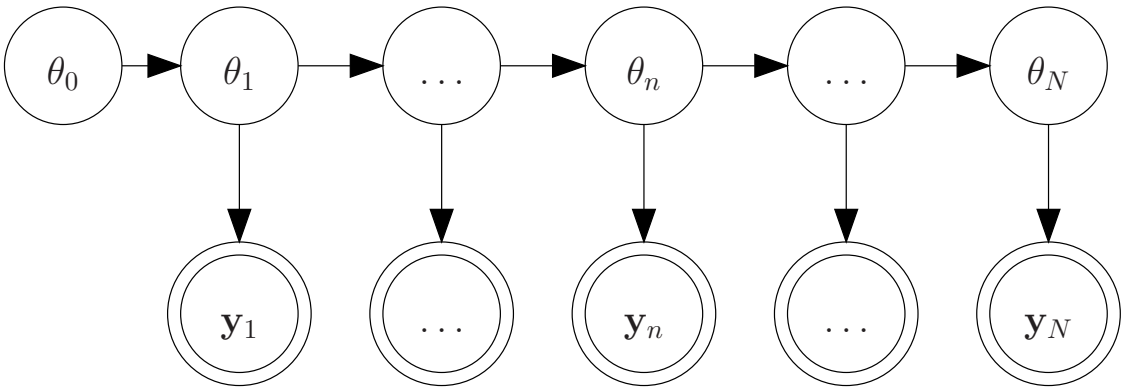


Figure 4.1: Hidden Markov model. Observed random variables have doubled lines

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | \mathrm{Par}(x_i)) \tag{4.4}$$

where $\mathrm{Par}(x_i)$ denotes the set of parent nodes of x_i. One weakness of Bayesian networks is that although conditionally dependencies are expressed directly, the task of determining whether a variable is conditionally dependent of another is not so clearly evident. We denote the *Markov blanket* of a node as the set of nodes for which, given the values of these nodes, the node is conditionally independent of all other nodes in the network. For a Bayesian network, the Markov blanket of x_i is the union set $\{\mathrm{Par}(x_i) \cup \mathrm{Chl}(x_i) \cup \mathrm{Par}(\mathrm{Chl}(x_i))\}$ where $\mathrm{Chl}(x_i)$ denotes the set of child nodes of x_i.

4.1.3 Hidden Markov Models

A hidden Markov model (HMM) is a probabilistic model with an underlying Markov process, that is, the conditional probability distribution of future states of the process given the present and all past states of the process is conditionally independent of the past states, i.e., it depends only on the present state. This state of the process is assumed to be hidden, and the task is to infer the sequence of states over time given some observations dependent only on the current value of the state. Usually the hidden Markov model is viewed as a special case of a general state space probabilistic model, where the state space Θ is discrete.

Hidden Markov models are often used for modelling dynamical systems. We will use the following notation, which is represented as a Bayesian network in Figure 4.1 on page 47: the hidden state sequence is $\theta_{0:K}$ and $y_{1:K}$ is the sequence of observations produced according to the state likelihoods (also known as *emission probabilities*) $p(y_k|\theta_k)$, over discrete times $k = 1, \ldots, K$. The state sequence evolves as a Markov chain, that is, it has an initial probability distribution $p(\theta_0)$ and a set of transition probability distributions $p(\theta_k|\theta_{1:k-1}) = p(\theta_k|\theta_{k-1}), k = 1, \ldots, K$.

4.1.4 Exponential Family of Probability Distributions

The exponential family of probability distributions is a class of probability distribution having the form

$$p(y|\theta) = \frac{1}{Z_\theta} e^{-\langle \theta, T(y) \rangle} \tag{4.5}$$

The normalizing factor Z_θ is given by $\int dy\, e^{-\langle \theta, T(y) \rangle}$. This family has a number of useful properties which make them important for Bayesian inference:

- The finite vector $T(y)$ is the collection of *sufficient statistics*, which capture all the possible information about θ that is represented by observations y, that is: $p(y|T(y), \theta) = p(y|T(y))\ \forall \theta$. For inference only the sufficient statistics $T(y)$ are required, not the entire data.
- The maximum likelihood parameters $\hat{\theta}_{ML}$ that maximize (4.5) are those for which the observed values of the sufficient statistics equal their expected values: $T(y) = \langle T \rangle_\theta$.
- For a likelihood function $p(y|\theta)$ in the exponential family, there exists a conjugate prior (4.1.1.5) $p(\theta)$, often itself in the exponential family, for which the posterior $p(\theta|y)$ is the same class of distribution as the prior. This is useful in variational methods (Section 4.2.2) as the update equations require only updating the sufficient statistics of a factor rather than performing a computationally intensive calculation over the whole parameter space of a probability distribution.

4.2 Inference Algorithms

In this section we cover inference algorithms falling into three main categories: exact inference (Section 4.2.1) for situations where it is possible to compute the posterior for all possible parameter settings, sampling methods (Section 4.2.2) which aim to generate samples from the posterior and compute expectations via Monte Carlo integration, and variational methods (Section 4.2.2) which approximate the full posterior with distributions for which the integrals can be computed.

4.2.1 Exact Inference

Exact inference refers to being able to compute posterior quantities precisely. Often this requires marginalizing over parameter spaces, which therefore requires the integrals to be analytic (such as in the case of models with Gaussian conditional probabilities) or that the parameters only assume discrete values. The Kalman filter is an exact inference algorithm for linear dynamical systems with Gaussian transition and observation noise. For a hidden Markov model (Section 4.1.3) with a discrete and finite state space, such that θ_k can assume one of E possible values, and the conditional probability of the observation given the state is computable, there exist a group of message-passing inference algorithms with complexity $\mathcal{O}(E^2 K)$ to compute various inference tasks.

Typically we may wish to determine the probability of states at time k, given past observations $p(\theta_k|y_{1:k})$, which is known as filtering and can be carried out recursively on-line, or including all future observations $p(\theta_k|y_{1:K})$ up to time K, which is known as smoothing and must be carried out offline, or including N recent observations $p(\theta_k|y_{1:k+N})$, which is known as fixed-lag smoothing and is practical if a certain amount of latency in the inference is acceptable. We may also wish to predict future states $p(\theta_{k+N}|y_{1:k})$ or infer the most likely sequence of states $p(\theta_{0:K}|y_{1:K})$ which is known as the *Viterbi* path. The computations required

for all these related but distinct queries can be viewed in terms of message passing algorithms. Both the Kalman filter and the following HMM algorithms are actually special cases of inference algorithms, known as the *sum-product* algorithm (for filtering and smoothing), that act over general Bayesian networks or factor graphs, see Bishop [2006] for details. Rabiner [1989] provides a tutorial on HMMs, also describing the Baum-Welch algorithm, which is an expectation-maximization (EM) algorithm for learning hidden parameters of the transition and observation process.

Let $\theta_{0:K}$ be the unknown state sequence in a hidden Markov model, and let $y_{1:K}$ be the sequence of observations generated. By Bayes' theorem, the posterior distribution over all possible state sequences is given by

$$p(\theta_{0:K}|y_{1:K}) = \frac{p(y_{1:K}|\theta_{0:K})p(\theta_{0:K})}{p(y_{1:K})} \tag{4.6}$$

The marginal filtering density $p(\theta_k|y_{1:k})$ can be computed up to the normalizing constant $p(y_{1:k})$ by passing $\alpha_{k|k}(\theta_k) \equiv p(\theta_k|y_{1:k})p(y_{1:k})$ 'alpha' messages between neighbouring frames:

$$\alpha_{0|0}(\theta_0) = p(\theta_0) \tag{4.7}$$

$$\alpha_{k|k-1}(\theta_k) = \sum_{\theta_{k-1}} p(\theta_k|\theta_{k-1})\alpha_{k-1|k-1}(\theta_{k-1}) \tag{4.8}$$

$$\alpha_{k|k}(\theta_k) = p(y_k|\theta_k)\alpha_{k|k-1}(\theta_k) \tag{4.9}$$

The marginal smoothing density $p(\theta_k|y_{1:K})$ is computed offline by passing $\beta_{k|k}(\theta_k) \equiv p(y_{k+1:K}|\theta_k)$ 'beta' messages as follows:

$$\beta_{K|K+1}(\theta_K) = 1 \tag{4.10}$$

$$\beta_{k|k}(\theta_k) = p(y_k|\theta_k)\beta_{k|k+1}(\theta_k) \tag{4.11}$$

$$\beta_{k|k+1}(\theta_k) = \sum_{\theta_{k+1}} p(\theta_{k+1}|\theta_k)\beta_{k+1|k+1}(\theta_{k+1}) \tag{4.12}$$

$$p(\theta_k|y_{1:K}) \propto \alpha_{k|k}(\theta_k)\,\beta_{k|k+1}(\theta_k) \tag{4.13}$$

The Viterbi path is computed in an analogous manner, where messages from neighbouring frames and observations are combined by taking the maximum rather than summing, i.e.,

$$\alpha_{k|k-1}(\theta_k) = \max_{\theta_{k-1}} p(\theta_k|\theta_{k-1})\alpha_{k-1|k-1}(\theta_{k-1}) \tag{4.14}$$

$$\beta_{k|k+1}(\theta_k) = \max_{\theta_{k+1}} p(\theta_{k+1}|\theta_k)\beta_{k+1|k+1}(\theta_{k+1}) \tag{4.15}$$

$$\arg\max_{\theta_{0:K}} p(\theta_{0:K}|y_{1:K}) = \arg\max_{k=0:K} \alpha_{k|k}(\theta_k)\,\beta_{k|k+1}(\theta_k) \tag{4.16}$$

Algorithm 4.1 Generic MCMC

- Sample $\tilde{\theta}^{(0)} \sim \pi_0(\theta)$
- For $i = 1, \ldots, M$
  - Sample $\tilde{\theta}^{(i)} \sim \mathcal{K}(\theta|\tilde{\theta}^{(i-1)})$

4.2.2 Monte Carlo Methods

Bayesian inference problems frequently involve computing high-dimensional integrals. For example, if we are interested in the mean of the posterior $p(\theta|y)$, we are required to compute the following integral over the space of possible parameter settings Θ.

$$E_{p(\theta|y)}[\theta] = \int_{\Theta} \theta p(\theta|y) d\theta \tag{4.17}$$

In many cases, the dimension of Θ is very large and the problem is intractable. We must resort to using a *Monte Carlo estimate*, which is a stochastic numerical integration method. Monte Carlo methods [Gilks and Spiegelhalter, 1996, Liu, 2003] are a general class of methods to compute expectations of random variables. We denote the target probability density function (pdf) of interest as $\pi(\theta)$, and assume we have a set of random samples $\tilde{\theta}^{(i)}, i = 1, \ldots, N$ drawn from $\pi(\theta)$, which are called *Monte Carlo samples*. Now, for a general function $h(\theta)$ over the parameter space, we have, by the law of large numbers, the following approximation to the general integration problem

$$\int_{\theta} h(\theta)\pi(\theta) d\theta \approx \frac{1}{N} \sum_{i=1}^{N} h(\tilde{\theta}^{(i)}) \tag{4.18}$$

$\pi(\theta)$ can however be of large dimension, and with an unknown normalizing constant, and therefore may be difficult to sample from. See Robert and Casella [2004] for a full overview of Monte Carlo techniques. Here we will mention two methods in particular which are useful to us for generating random samples from such a pdf.

4.2.2.1 Markov Chain Monte Carlo

Markov Chain Monte-Carlo (MCMC) methods construct a Markov chain with stationary distribution equal to the target pdf $\pi(\theta)$ which we wish to sample from. A Markov chain is specified by an initial sample distribution $\pi_0(\theta)$ and a transition kernel $\mathcal{K}(\theta|\theta')$ which is a probability density. For the stationary distribution of the Markov chain to be equal to the target pdf $\pi(\theta)$ the transition kernel must obey

$$\int_{\Theta} \mathcal{K}(\theta|\theta')\pi(\theta') d\theta' = \pi(\theta) \quad \forall \theta \in \Theta \tag{4.19}$$

Given such a kernel, the Monte Carlo samples $\tilde{\theta}^{(i)}$ are generated from the Markov chain as in 4.1.

A general method of constructing the kernel $\mathcal{K}(\theta|\theta')$ is provided by the Metropolis-Hastings (MH) algorithm. Suppose we have a *proposal pdf* $q(\theta|\theta')$ which we can directly sample from, with $q(\theta|\theta') > 0$ wherever $\pi(\theta) > 0$. Step 3 in 4.1, which involves drawing samples from the kernel, is expanded in 4.2, where candidate

Algorithm 4.2 Metropolis-Hastings Kernel

- Sample $\theta^\star \sim q(\theta|\tilde{\theta}^{(i-1)})$
- Compute the acceptance ratio $\alpha_{\text{MH}} = \min\left[1, \frac{\pi(\theta^\star)}{\pi(\tilde{\theta}^{(i-1)})} \frac{q(\tilde{\theta}^{(i-1)}|\theta^\star)}{q(\theta^\star|\tilde{\theta}^{(i-1)})}\right]$
- *Accept*: with probability α_{MH} set $\tilde{\theta}^{(i)} \leftarrow \theta^\star$
- *Reject*: otherwise set $\tilde{\theta}^{(i)} \leftarrow \tilde{\theta}^{(i-1)}$

samples $\theta^\star$ are drawn from the proposal pdf $q(\theta|\theta')$, and then accepted or rejected according to an acceptance ratio or probability α_{MH}.

It is not necessary for the entire parameter θ to be sampled in each step. Let θ be partitioned into D disjoint components $\{\theta_1, \theta_2, \ldots, \theta_D\}$ which may be groups (*blocks*) of parameters. It is permissible at each iteration for the target pdf to be one of $p(\theta_d|\theta_{\neg d})$, $d = 1, \ldots, D$ provided every component θ_d has some probability of being chosen at each iteration. For the special case when we are able to sample directly from $p(\theta_d|\theta_{\neg d})$ then $\alpha_{\text{MH}} = 1$, that is, every candidate is accepted. This is known as *Gibb's sampler*. This can typical occur in Bayesian networks with conditional dependencies between the nodes defined as standard probability distributions. The algorithm can then be viewed in terms of passing messages between nodes, and has been implemented in software packages such as WinBugs and OpenBugs.

Commonly the Markov chain is allowed to run for a *burn-in* period before computing statistics of the Monte Carlo samples, as it is a commonly observed phenomenon that the Markov chain 'converges' to likely parameter settings after such a period.

[Green, 1995] extends MCMC to model selection problems, where the model parameters have different dimensions. The scheme is known as Reversible Jump MCMC. An MH kernel is required to move between models M_j and $M_{j'}$ of differing dimension. The jump must be reversible in that for every accepted move $\theta_j \to \theta_{j'}$, the reverse move $\theta_{j'} \to \theta_j$ must have positive probability of acceptance. The MH proposals are normally designed so that θ_j and $\theta_{j'}$ share most of the parameter settings between them, and therefore a sensible proposal normally requires that the difference between the dimensions of j and j' is small.

The MH proposal pdf is augmented to become $q(\theta_{j'}, u_{j'}|\theta_j, u_j)$ where $u_{j'}$ and u_j are auxiliary random variables that keep the dimensions between the augmented parameter spaces constant: $D(\theta_{j'}) + D(u_{j'}) = D(\theta_j) + D(u_j)$ where $D(\cdot)$ is the dimension of the random variable. The acceptance probability in 4.2 is replaced by

$$\alpha_{MH} = \min\left[1, \frac{\pi(\theta_j^\star)}{\pi(\tilde{\theta}_j)} \frac{q(\theta_j^\star, u_j|\tilde{\theta}_j, u_j)}{q(\tilde{\theta}_j, u_j|\theta_j^\star, u_j)} \left|\frac{\partial(\theta_j^\star, u_j)}{\partial(\tilde{\theta}_j, u_j)}\right|\right] \tag{4.20}$$

4.2.2.2 Importance Sampling and Sequential Monte Carlo

The Metropolis-Hastings algorithm derives from acceptance-rejection sampling. Another sampling method, importance-sampling, leads to a class of algorithms known as *sequential Monte Carlo*, so called because they are often suitable for sequential inference problems as often appear in dynamical systems and hidden Markov models. Importance sampling requires another proposal pdf $q(\theta)$ with $q(\theta|\theta') > 0$ wherever $\pi(\theta) > 0$, known as the *importance pdf*. Now, if we have a set of Monte Carlo samples $\tilde{\theta}^{(i)}, i = 1, \ldots, N$ drawn from $q(\theta)$, we

can write the general integration problem as

$$\int_\theta h(\theta)\pi(\theta)d\theta = \int_\theta h(\theta)\frac{\pi(\theta)}{q(\theta)}q(\theta)d\theta \approx \frac{1}{N}\sum_{i=1}^{N}\tilde{\omega}^{(i)}h(\tilde{\theta}^{(i)}) \tag{4.21}$$

Each Monte Carlo sample $\tilde{\theta}^{(i)}$ is weighted by an *importance weight* $\tilde{\omega}^{(i)}$ which corrects the difference between $q(\theta)$ and $\pi(\theta)$. The importance weights are given by the importance ratio

$$\tilde{\omega}^{(i)} = \frac{\pi(\tilde{\theta}^{(i)})}{q(\tilde{\theta}^{(i)})} \tag{4.22}$$

The variance of the estimate depends strongly on how close $q(\theta)$ is to $\pi(\theta)$, which is usually evaluated as the Kullback-Liebler divergence $KL(q(\theta)||\pi(\theta))$. The optimal importance pdf that minimizes the variance of the estimate is $q(\theta) = |h(\theta)|\,\pi(\theta)$, which however usually cannot be sampled from.

Sequential Monte Carlo methods [Doucet et al., 2001], also known as particle filtering, are algorithms for approximate inference on typically dynamical systems using importance sampling. At each time k we are interested in generating Monte Carlo samples $\theta_{1:k}^{(i)}$ of the state trajectories, also know as *particles*. The target posterior pdf $p(\theta_{1:k}|y_{1:k})$ can be written sequentially as

$$p(\theta_{1:k}|y_{1:k}) = p(\theta_{1:k-1}|y_{1:k-1})\frac{p(\theta_k|\theta_{k-1})p(y_k|\theta_k)}{p(y_k|y_{1:k-1})} \tag{4.23}$$

The normalization constant $p(y_k|y_{1:k-1})$ does not need to be computed. We also select a sequential importance pdf

$$q_k(\theta_{1:k}) = q_{k-1}(\theta_{1:k-1})q_k(\theta_k|\theta_{k-1}) = q_0(\theta_0)\prod_{k'=1}^{k} q_{k'}(\theta_{k'}|\theta_{k'-1}) \tag{4.24}$$

so we can sample the new state at time k from the proposal $q_k(\theta_k|\theta_{k-1})$ using the state trajectories found up until time $k-1$. The choice of the proposal pdf affects the resulting variance of the particle estimate. The optimal proposal uses the new observation: $q_k(\theta_k|\theta_{k-1}) = p(\theta_k|\theta_{k-1}, y_n)$ but can be impossible to sample from. Local Gaussian approximations to the optimal proposal lead to the unscented and extended Kalman filters. The simplest proposal is the transition pdf $q_k(\theta_k|\theta_{k-1}) = p(\theta_k|\theta_{k-1})$ which results in the algorithm called the bootstrap filter (4.3).

The importance weight of each particle is given as the ratio

$$\frac{p(\theta_{1:k}|y_{1:k})}{q_k(\theta_{1:k})} \tag{4.26}$$

However, after a few iterations, most of these weights become close to zero, and the solution is degenerate, being represented by only a few particles. The solution is to resample the particles: particles with low weights are moved to more accurate positions. A resampling step takes place when the efficiency number $I_{\text{eff}} = [\sum_{i=1}^{I}(\tilde{w}_k^{(i)})^2]^{-1}$ is lower than some threshold. A simple scheme known as stratified sampling samples such that the expected number of particles following the resampling step at θ_k is equal to $I\tilde{w}_k^{(i)}$. See Doucet et al. [2001] for further details and alternative resampling schemes.

Algorithm 4.3 Bootstrap Particle Filter

- Initialize $\tilde{\theta}_0^{(i)} \sim q_0(\theta_0)$, $\tilde{w}_0^{(i)} = p_0(\tilde{\theta}_0)/q_0(\tilde{\theta}_0)$ for each particle
- For $k = 1, \ldots, K$
  - Update particle trajectories $\tilde{\theta}_k^{(i)} \sim q_k(\theta_k|\tilde{\theta}_{k-1}^{(i)})$
  - Compute particle weights

$$\tilde{w}_k^{(i)} \propto \tilde{w}_{k-1}^{(i)} \frac{p(\tilde{\theta}_k^{(i)}|\tilde{\theta}_{k-1}^{(i)}) p(y_k|\tilde{\theta}_k^{(i)})}{q_k(\tilde{\theta}_k^{(i)}|\tilde{\theta}_{k-1}^{(i)})} \tag{4.25}$$

    and normalize: $\tilde{w}_k^{(i)} \leftarrow \tilde{w}_k^{(i)} / \sum_{i'=1}^{I} \tilde{w}_k^{(i')}$
  - Resample if necessary

4.2.3 Variational Methods

Variational methods are an alternative, deterministic method for making approximate posterior estimates. Define $p(y, \theta)$ as the joint distribution when y is observed, with normalizing constant Z_y. We wish to compute the posterior

$$p(\theta|y) = \frac{1}{Z_y} p(y, \theta) \tag{4.27}$$

$$Z_y = \int p(y, \theta) d\theta \tag{4.28}$$

The Monte Carlo techniques described in Section 4.2.1 can be utilized to draw samples from $p(y, \theta)/Z_y$. However, in practice, the integral can often be more quickly approximated by the *structured mean field* method, also known as *variational Bayes*. The integrand $\mathcal{P} = p(y, \theta)/Z_y$ is approximated with a simpler distribution $\mathcal{Q}$ such that the integral in (4.28) is tractable. A common factorization involves partitioning the parameters into d disjoint components, $\theta_1, \ldots, \theta_D$ such that

$$\mathcal{Q} = \prod_{d=1}^{D} \mathcal{Q}_d(\theta_d) \tag{4.29}$$

The mean field method minimizes the KL divergence $KL(\mathcal{P}||\mathcal{Q})$. Due to the non-negativity of the KL measure, we then obtain a lower bound on the normalizing constant

$$\log Z_y \geq \langle \log p(y, \theta) \rangle_{\mathcal{Q}} - \langle \log \mathcal{Q} \rangle_{\mathcal{Q}} \tag{4.30}$$

The second term of (4.30) is the entropy: $-H[\mathcal{Q}] \equiv \langle \log \mathcal{Q} \rangle_{\mathcal{Q}}$. See Chapter A for expressions for the entropy of probability distributions in the exponential family. The factors $\mathcal{Q}_d$ obey the fixed point equation

$$\mathcal{Q}_d \propto \exp(\langle \log p(y, \theta) \rangle_{\mathcal{Q}_{\neg d}}) \tag{4.31}$$

which can be computed easily if all the factor distributions are chosen to be in a conjugate-exponential

family. For example, if a random variable σ^2 has sufficient statistics $\left\langle 1/\sigma^2 \right\rangle_{\mathcal{IG}}$ and $\left\langle \log \sigma^2 \right\rangle_{\mathcal{IG}}$ under an Inverse-Gamma distribution, and $y \sim \mathcal{N}(0, \sigma^2)$ then

$$\mathcal{Q}_y \propto \exp\left(-\tfrac{1}{2}\left\langle 1/\sigma^2 \right\rangle y^2 - \tfrac{1}{2}\log 2\pi - \tfrac{1}{2}\left\langle \log \sigma^2 \right\rangle\right) \tag{4.32}$$

and the VB update is $\langle y \rangle_{\mathcal{N}} = 0$, $\left\langle y^2 \right\rangle_{\mathcal{N}} = \left\langle 1/\sigma^2 \right\rangle^{-1}$ as expected. See Chapter A for expressions of the sufficient statistics of some probability distributions in the exponential family. The fixed point equation (4.31) has the property that for every iteration, the lower bound (4.30) is guaranteed to increase.

Variational Bayes and Gibbs' sampler have been compared for inference in audio signal models in Godsill et al. [2007], Cemgil et al. [2007]. Qualitatively, VB methods tend to converge quicker than Gibbs' sampler, but may result in a poorer solution because only a lower bound of the likelihood is being computed.

Chapter 5

A Signal Model for Pitched Musical Instruments

We consider the modelling of pitched musical instruments as a summation of several sinusoids with correlated frequency, amplitude and phase. Background noise and modelling error are treated as Gaussian noise, which makes a probabilistic treatment desirable. We introduce appropriate priors for the model parameters, which are chosen to reflect prior knowledge of the structure of pitched musical note signals, and to allow effective numerical Bayesian inference. The model we introduce is shown to be capable of modelling both frequency and amplitude modulations, which are characteristic of the sound of many musical instruments. The use of a Bayesian methodology allows model selection to be carried out implicitly, so that the relevant number of sinusoids necessary to model the signal appropriately may be determined automatically.

5.1 Contributions

The motivation for this chapter is to extend and further develop promising Bayesian generative models based on a sinusoidal representation for each partial frequency, and present the developments in such a way that they can be incorporated easily into and compared with existing approaches.

In 5.2.2 we formulate the mathematical representation of a sinusoid for which the amplitude and phase are permitted to vary slowly in comparison to the central frequency. This motivates the use of the analytic representation in 5.2.3, which eliminates ambiguity in the sinusoidal representation, and we show that this analytic representation is appropriate to be applied to existing Bayesian models using sinusoidal representations. We then describe a state-space representation with constant damping ratio in 5.2.4, showing that the analytic representation can result in a closed form posterior distribution for the damping ratio and frequency, which does not typically arise in the literature, and may be used to linearize the posterior distribution under arbitrary choices of frequency priors. In 5.2.5 we apply the analytic representation to Gabor models, and additionally motivate the use of sinc basis functions to specify and control the bandwidth of frequency and amplitude modulations in the signal. The use of sinc basis functions may be incorporated into the existing methods independently of whether the analytic representation is used or not.

In Section 5.3 we describe the literature for Bayesian inference in sinusoidal models and a noise model

that may be used. We then derive MCMC algorithms for a fixed number of partials with an arbitrary prior on partial frequencies for both the state-space and Gabor models. For the state-space model in particular, we derive the posterior distribution of the frequencies and damping ratios under a normally distributed prior, and show how this can be adapted as an efficient, model-based, proposal distribution when the prior is not a normal distribution.

5.2 Model for an Isolated Partial

In this section we consider how to model the signal of an isolated partial frequency. The focus on this section is to introduce various representations of a sinusoidal signal and how the amplitude envelope and modulations around the central frequency can be handled. In the following section, we will consider the superposition of multiple sinusoids and embed the entire model in a probabilistic framework for Bayesian inference.

5.2.1 Motivation

An isolated partial is a minimal description of a musical note. When we listen to an isolated partial, we have the clear perception of the pitch and volume of a musical note. The pitch is related to the frequency of the partial, although the perception of pitch itself is non-linear (2.2.2). Some degree of frequency modulation is tolerated, being perceived as vibrato. The timbre of an isolated partial is perceived as the purest tone, as there is no harmonic structure and therefore no possibility of inharmonicity. Musical instruments which may be modelled by isolated partials include whistles, tuning forks and rubbing crystal glasses.

The model we introduce is parametric in that the sinusoid is completely described by its frequency, phase and amplitude envelope; and linear so that we may superimpose multiple sinusoids in order to generate more complex musical tones. We require a comprehensive understanding of the parameters of even such a simple model, because we will pursue the intuition that the perceptual grouping of partials into musical notes (rather than being perceived as separate frequencies) is due to the group of partials having a shared set of parameters.

5.2.2 Amplitude and Phase Modulation

In this chapter we will consider a segment of audio data with N samples and time indices $t = 0, \ldots, N-1$, where it is assumed that a set of multiple pitches are sounding throughout the length of the segment. We will begin by modelling an isolated partial as a sinusoid $x[t]$ having a constant angular frequency ω, amplitude envelope $c[t]$ and time-varying phase $\phi[t]$:

$$x[t] = c[t] \cos[\omega t + \phi[t]] \tag{5.1}$$

For this model to be realistic, we require constraints on the amplitude envelope and phase modulation. The amplitude envelope is used to model changes in the perceived volume of the partial. Hence the bandwidth of the envelope should be restricted to the lower limit of hearing (20Hz), otherwise the frequency content of the envelope will be perceived as an additional pitch. The ear is relatively insensitive to the phase of a pitched

note, but phase modulations may be perceived as frequency modulations, as the modulation in frequency around ω is given by the time derivative of $\phi[t]$. Vibrato (see 2.3.4) is common in many musical genres and instruments, and results in both frequency and amplitude modulations of the note. Experimental studies by Brown and Vaughn [1996] have shown that the perceived pitch centre of a vibrato note is still equivalent to the centre frequency ω. The permissible amount of frequency modulation is governed by stylistic rules, however a useful guideline is that the depth of vibrato should not cross the frequency boundary of the pitch, which in Western music is a semitone. The speed and depth of vibrato are also limited by the mechanical process which creates the vibrato effect, for example in the case of a violin, the rocking of the violinist's finger on the string.

5.2.3 Analytic Representation of Sinusoidal and Noise Signals

In (5.1) there is ambiguity in the definitions of $c[t]$ and $\phi[t]$. Different choices will result in the same identical signal $x[t]$. To overcome this Gabor [1946] defines the instantaneous amplitude and instantaneous phase using the *analytic representation* of a signal. This has become the conventional definition, and reduces the ambiguity (see Cohen et al. [1999] for cases where the ambiguity still exists). An analytic signal is a complex valued signal with no negative frequency components in its Fourier spectrum. The analytic representation of a real valued signal is produced by discarding the negative frequency components of the Fourier transform. There is no loss of information as the Fourier spectrum of a real signal has Hermitian symmetry around zero frequency.

The analytic representation $x_a[t]$ of a real-valued signal $x[t]$is given by

$$x_a[t] \equiv x[t] + i\mathcal{H}[x[t]] \tag{5.2}$$

where $\mathcal{H}$ denotes the Hilbert transform. The Hilbert transform shifts the phase of negative frequency components of the Fourier spectrum by $+\pi/2$ and the positive frequency components by $-\pi/2$. Thus the operation in (5.2) discards the negative frequency components. The original signal may be simply recovered from the real part of the analytic signal:

$$x[t] = \mathcal{R}(x_a[t])$$

The instantaneous amplitude of the analytic representation is defined as $|x_a[t]|$ and the instantaneous phase is defined as $\arg x_a[t]$. For the model of the isolated sinusoid (5.1) we have

$$\begin{aligned} x_a[t] &= c[t]\cos[\omega t + \phi[t]] + ic[t]\sin[\omega t + \phi[t]] \\ &= c[t]\exp i[\omega t + \phi[t]] \end{aligned}$$

from which we can see that the instantaneous amplitude is $c[t]$ and the instantaneous phase is given by $\omega t + \phi[t]$.

We will find it convenient to use the analytic representation for our models, and will particularly use the following form

$$x_a[t] = c[t]\exp[i\phi[t]]\exp[i\omega t] \tag{5.3}$$

as the three parameters of the sinusoid are thus separated. The analytic signal is composed of three separate

signals multiplied (modulated) together. Using communications terminology, $c[t]$ is the amplitude waveform, $\exp[i\phi[t]]$ is the phase waveform, and $\exp[i\omega t]$ is the carrier signal with frequency ω. To be able to transmit and recover the original waveforms from the modulated signal, it is necessary for the bandwidths of the amplitude and phase waveforms to be much smaller than ω. We will use the same concept for modelling musical signals, as extracting and storing these low bandwidth waveforms is attractive for compression, reconstruction and synthesis.

The model which we develop in this chapter is based on existing work on methods for sinusoidal models [Serra, 1997, Walmsley et al., 1999, Davy and Godsill, 2003, etc.], and it is necessary for us to confirm that the properties of these models are consistent with the analytic representation of the signal. The Hilbert transform is a linear operator, so the frequencies and amplitudes of the sinusoids are preserved. Moreover Picinbono and Bondon [1997] show that the analytic representation of a wide-sense stationary real signal is proper or circular symmetric [Neeser and Massey, 1993]. Hence the analytic representation of a white noise process is a complex Gaussian random variable, and the properties of an autoregressive (AR) process used to model coloured noise are retained in the analytic representation.

In 5.2.4 and 5.2.5, we consider two common formulations of the sinusoidal signal (5.3) which are used in state-of-the-art Bayesian harmonic models. Our contribution here is to apply these formulations to the analytic representation of the signal, and demonstrating how bandwidth constraints on frequency and amplitude modulations may be naturally and practically applied.

5.2.4 State-Space Formulation

The first formulation treats the sinusoid as a rotating phasor, and is motivated the work of Cemgil et al. [2006] who use a state-space approach to model the rotation of a real-valued sinusoid from one sample to the next. This approach was used in a polyphonic transcription system capable of resolving note onsets and offsets to sample resolution. Moreover as the notes are processed sample by sample and not on a frame by frame basis there are no artifacts arising from reconstruction and synthesis due to phase discontinuities and discrepancies at frame boundaries.

In our model, the relationship between one sample of the sinusoid and the next is given by

$$\begin{aligned} x_a[t+1] &= c[t+1]\exp[i\phi[t+1]]\exp[i\omega(t+1)] \\ &= c[t+1]\exp[i\phi[t+1]]\exp[i\omega t]\exp[i\omega] \\ &= \frac{c[t+1]}{c[t]}\frac{\exp[i\phi[t+1]]}{\exp[i\phi[t]]}\exp[i\omega]\, c[t]\exp[i\phi[t]]\exp[i\omega t] \\ &= \frac{c[t+1]}{c[t]}\exp[i\phi[t+1]-i\phi[t]]\exp[i\omega]\, x_a[t] \end{aligned} \tag{5.4}$$

The $\frac{c[t+1]}{c[t]}$ term in (5.4) gives the rise or decay in the amplitude envelope between t and $t+1$. Following Cemgil et al. [2006] we refer to this as the damping ratio, and define $\rho[t] \equiv \frac{c[t+1]}{c[t]}$. We will apply a constraint on the amplitude envelope by choosing to make this damping ratio constant throughout the segment of audio: $\rho[t] = \rho$ for all $t = 0, \ldots, N-1$. This is appropriate for musical instruments such as the piano and guitar, where after the onset of the note (when the string is struck by a hammer or plucked) the decay of the energy in each partial can be approximately described as exponential, $0 < \rho < 1$. It is also appropriate

for notes held at a constant volume, where $\rho = 1$. For other situations where other shapes of amplitude envelope would be expected, the Gabor model described in the next section is more appropriate.

The $\exp[i\phi[t+1] - i\phi[t]]$ term in (5.4) is the difference in the phase modulations, which can be seen as an approximation to the frequency modulation at t. Realistic frequency modulations should be small, hence we approximate using the Taylor expansion:

$$\exp[i\phi[t+1] - i\phi[t]] \approx 1 + (i\phi[t+1] - i\phi[t]) \tag{5.5}$$

The second term in (5.5) which we will denote as $f[t] \equiv i\phi[t+1] - i\phi[t]$ is small, and is purely imaginary, as the phase $\phi[t]$ is real for all t. However for convenience, we will model this frequency modulation term as a zero mean complex Gaussian random variable with small variance σ_f^2, i.e.,

$$p(f[t]) = \mathcal{N}_C\left(0, \sigma_f^2\right)$$

The consequence of $f[t]$ having a real part is that small amplitude modulations in addition to the damping ratio ρ are permitted. As stated in 5.2.2, frequency modulations in a musical note are often accompanied by amplitude modulations, hence we do not consider this inconsistency in our model a disadvantage.

When we incorporate the above constraints into (5.4) we have

$$x_a[t+1] = \rho \exp[i\omega] x_a[t] + f[t] \tag{5.6}$$

or alternative expressed as a conditional probability distribution:

$$p\left(x_a[t+1] | x_a[t], \rho, \exp[i\omega], \sigma_f^2\right) = \mathcal{N}_C\left(\rho \exp[i\omega] x_a[t], \sigma_f^2\right) \tag{5.7}$$

(5.6) and (5.7) show us that $x_a[t]$ can be regarded as the internal state of a linear dynamic system. This fact was used in Cemgil et al. [2006] where the parameters ρ, ω and σ_f^2 were known, and the Kalman filter used to infer the sinusoid in the presence of observation noise. In 5.3.3 we will show that these parameters may be treated as unknown and Bayesian inference can be used to estimate them.

5.2.5 Gabor Model

A model for slowly varying partial amplitudes was introduced by Godsill and Davy [2002] as an extension of the existing harmonic model of Walmsley et al. [1999] which assumed that the amplitude of each partial is constant throughout the note segment. Each partial is projected onto a set of Gabor functions $\psi_{i,\omega}[t - i\Delta]$, each of which has a fixed real-valued envelope $\psi[t]$, symmetric around $t = 0$ and having a finite region of support, shifted in time by $i\Delta$ and modulated by frequency ω equal to the frequency of the partial:

$$\psi_{i,\omega}[t] = \psi[t - i\Delta] \exp[i\omega t]$$

The constant Δ is the difference, in samples, between the centres of adjacent basis functions and controls the spacing along the time axis between neighbouring Gabor atoms. Δ is chosen such that the support of each function overlaps with the next, thus ensuring that the amplitude envelope varies smoothly throughout

the length of the note segment.

The Gabor model applied to the analytic representation of a signal is the projection of $x_a[t]$ onto $I+1$ Gabor functions, where $\Delta(I+1)$ is equal to the length N of $x_a[t]$:

$$x_a[t] = \sum_{i=0}^{I} b_i \psi[t - i\Delta] \exp[i\omega t] \tag{5.8}$$

The complex-valued basis coefficients b_i may be viewed as the amplitude of each Gabor function $\psi[t - i\Delta] \exp[i\omega t]$.

The amplitude envelope as modelled, comparing with (5.3), is given by

$$c[t] \exp[i\phi[t]] = \sum_{i=-\infty}^{\infty} b_i \psi[t - i\Delta] \tag{5.9}$$

and is also used to account for frequency modulations in our model. From a signal processing perspective, the envelope would be obtained by low-pass filtering the partial to remove the frequency component at ω of the spectrum. If we were to select the envelope of the Gabor function as the sinc function

$$\psi[t] = \frac{\sin[2\pi t/\Delta]}{2\pi t/\Delta} \tag{5.10}$$

then the bandwidth of the amplitude envelope (5.9) is constrained to $1/\Delta$. This result is based on the use of the sinc filter for perfect reconstruction of bandlimited signals [Shannon, 1998].

Godsill and Davy [2002] use a Hamming window as the envelope for the Gabor basis functions. When we use a sinc function (5.10) to model sinusoids with periodic amplitude and frequency modulations embedded in white noise, we have found that the residual of the modelling is smaller than when using Hanning windows, and the reconstruction sounds better. Figure 5.1 on page 61 compares the reconstructions obtained of a sinusoid by sinc and Hamming basis functions. The sinusoid has a central frequency of 440Hz, with a frequency modulation of depth 5Hz and speed 5Hz, and amplitude modulation of magnitude 0.2 and speed 5Hz. Ten basis functions were used to cover the entire signal length of 1 second, hence modulations up to 10Hz can be captured. Both basis functions model the spectrum well around the central frequency, but the sinc basis model has a smaller residual and fewer reconstruction artefacts away from the central frequency.

In practice we limit the support of the sinc function to 4Δ i.e.,

$$\psi[t] = \begin{cases} \frac{\sin[2\pi t/\Delta]}{2\pi t/\Delta} & |t| \leq 4\Delta \\ 0 & |t| > 4\Delta \end{cases}$$

as the amplitude of the envelope is small outside the central region.

5.3 Probabilistic Model for Multiple Partials

In this section we will combine multiple instances of the models for isolated partials described in 5.2.2 and embed them in observation noise. The combined model may then be used for estimating the spectrum of musical signals. We will adopt a Bayesian approach throughout, and our goal is to jointly infer the number

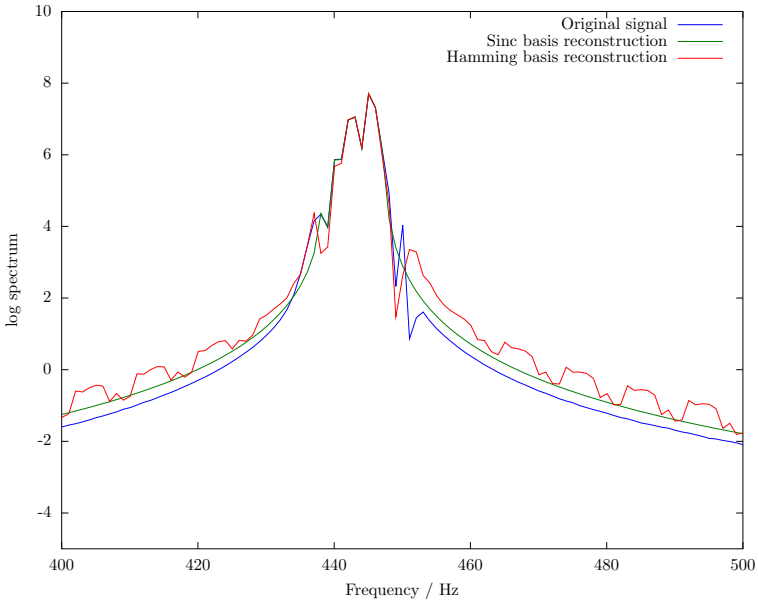


Figure 5.1: Comparison of the reconstructions of a violin note with fundamental frequency 440Hz and played with vibrato, using sinc and Hamming basis functions. The sinc basis reconstruction has a smooth spectral shape matching the original signal, whereas the reconstruction using the Hamming basis has a periodic artefact resulting from difficulties modelling the frequency and amplitude modulations in the signal.

of partials and their frequency and amplitudes through Bayesian model selection.

5.3.1 Background

Full Bayesian inference of a sinusoidal model with noise was first carried out by Andrieu and Doucet [1999] using a reversible jump MCMC scheme (4.2.2.1). A short frame of samples is modelled by a set of constant amplitude sinusoids in white noise, and the inference scheme is shown to correctly and robustly estimate the number of sinusoids present even at low signal-to-noise ratios. The conditional distribution of the frequencies of the sinusoids is not however of a form which can be sampled easily. By this we mean that $p(\omega|y,\theta)$ where ω is the set of sinusoid frequencies, y is the observed data, and θ are the remainder of the model parameters, is not of a standard form for which a sampling algorithm is known. Two Metropolis-Hastings proposal schemes are suggested for updating the frequencies of the sinusoids from $\omega^{(i)}$ in iteration i of the algorithm, to $\omega^{(i+1)}$. The first is a *local* proposal which generates candidates ω' from a Gaussian distribution with mean $\omega^{(i)}$ and small variance. This allows the frequencies to be estimated to a high precision. The second is a *global* proposal which generates candidates ω' independently from $\omega^{(i)}$, with probability proportional to the Fourier spectrum. This allows the Markov chain to explore new regions of the spectrum. Andrieu and Doucet [1999] provide the probabilities at which to accept each proposal, and also describe birth and death moves for sinusoids in the reversible jump framework, so that the number of sinusoids can be estimated.

Walmsley et al. [1999] extend the above model to harmonic signals, where the frequencies of each partial are set to integer multiples of the fundamental frequency. Notes in the model are turned on and off using

binary indicator variables. The global proposal is facilitated by a harmonic transform which functions in a similar way to a comb filter (2.2.2), incorporating the energy of the higher-order harmonics into the fundamental and low-order harmonics. An additional proposal is designed to allow the inference algorithm to explore octave errors.

Godsill and Davy [2002] further extend the model so that each sinusoid is modelled by a set of Gabor basis functions, as outlined in 5.2.5. Inharmonicity is introduced into the model, originally as an additive term, then as a multiplicative term in Godsill and Davy [2005] (3.3). Reversible jump MCMC is again used, and a range of moves are proposed to explore the high dimensional model space fully: note births and deaths, adding and subtracting variable numbers of harmonics from each note, and multiplying or dividing the fundamental frequency by a factor of two to explore octave errors.

Cemgil et al. [2006] use the state-space representation outlined in 5.2.4 in a polyphonic transcription system. The partial frequencies are fixed to MIDI specification frequencies (2.3) and the amplitude envelopes of the notes are fixed. The note onsets and offsets are inferred using a pruning algorithm.

In this section we will use much of the prior structure that has been developed by the above authors, and apply it to the analytical representation of the signal with the constraints on the amplitude and frequency modulation as described in Section 5.2. The contributions made in this section are improvements to and developments of the state-of-the-art inference algorithms for these model. For the state-space model, the posterior distribution of the frequency parameters under a normal distribution prior is available in closed-form, and this fact is used to derive a Gibbs sampler for the normal distribution prior. For an arbitrary prior distribution on the frequency parameters a Metropolis-Hastings MCMC algorithm is derived using a linearization of the posterior distribution as an efficient proposal distribution. This allows the state-space representation to be used to accurately infer frequencies using a rich prior model for inharmonicity, as will be demonstrated in Section 5.5. Prior to this work, inference on the state-space model was restricted to a fixed grid of frequencies [Cemgil et al., 2006]. For the Gabor model, the contribution is the derivation of the posterior mode of a signal-to-noise ratio hyperparameter, allowing this parameter to be inferred from a marginalized distribution, thus improving estimation and eliminating the computation required simulating latent parameters.

5.3.2 Noise Model

In this chapter we have chosen to use a white noise model. As the sinusoidal models we consider here are linear, it is straightforward to model coloured noise sources using an autoregressive (AR) process. For the state-space model, the extension required is straightforward as the AR model is itself commonly expressed as a state-space model. For the extensions required to the Gabor model, see Godsill and Davy [2002].

For the remainder of this chapter, we drop the subscript a denoting that $x_a[t]$ is an analytic representation, and work with M partials which we denote $x_m[t]$. The white noise process is denoted $n[t]$ and has variance σ_n^2. The signal we observe is denoted $y[t]$ and is given by

$$y[t] = \sum_{m=1}^{M} x_m[t] + n[t] \tag{5.11}$$

We choose the prior distribution of σ_n^2 to be inverse-Gamma

$$p\left(\sigma_n^2\right) = \mathcal{IG}\left(\sigma_n^2; \alpha_n, \beta_n\right)$$

such that the conditional distribution of σ_n^2, given $\mathbf{n} \equiv [n[0], \ldots, n[N-1]]^\top$, is

$$p\left(\sigma_n^2|\mathbf{n}\right) = \mathcal{IG}\left(\sigma_n^2; \alpha_n + \frac{N}{2}, \beta_n + \frac{1}{2}\mathbf{n}^\top\mathbf{n}\right) \tag{5.12}$$

A common setting is $\alpha_n = \beta_n = 0$ such that

$$p\left(\sigma_n^2\right) \propto \frac{1}{\sigma_n^2}$$

This prior is invariant to arbitrary scaling of the observed signal, and has a maximum entropy interpretation [Jeffreys, 1946].

The structure of the signal model depends on the parametrization that we have chosen.

5.3.3 State-Space Formulation

For the state-space formulation, each partial $x_m[t]$ has an unknown damping ratio ρ_m, and frequency ω_m. From (5.7) and (5.11) the model is

$$\begin{aligned} p\left(x_m[t+1]|x_m[t], \rho_m, \omega_m, \sigma_f^2\right) &= \mathcal{N}_C\left(x_m[t+1]; \rho_m \exp[i\omega_m] x_m[t], \sigma_f^2\right) \\ p\left(y[t]|x_1[t], \ldots, x_M[t], M, \sigma_n^2\right) &= \mathcal{N}_C\left(y[t]; \sum_{m=1}^{M} x_m[t], \sigma_n^2\right) \end{aligned} \tag{5.13}$$

(5.13) is a linear dynamical system, with an unobserved state vector $\mathbf{x}_t = [x_1[t], \ldots, x_M[t]]^\top$ at time t, diagonal $M \times M$ state transition matrix $\mathbf{A}$ with elements $\rho_m \exp[i\omega_m]$ along the diagonal, process noise covariance matrix $\sigma_f^2\mathbf{I}_M$, observation model $\mathbf{H}$ which is a $1 \times M$ vector with all elements equal to one, and observation noise variance σ_n^2:

$$\begin{aligned} p\left(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{A}, \sigma_f^2\right) &= \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{A}\mathbf{x}_t, \sigma_f^2\mathbf{I}_M\right) \\ p\left(y[t]|\mathbf{x}_t, \sigma_n^2\right) &= \mathcal{N}\left(y[t]; \mathbf{H}\mathbf{x}_t, \sigma_n^2\right) \end{aligned} \tag{5.14}$$

This is in a standard form for inferring the marginal distribution of the state vector $\mathbf{x}_t$ at each time t given the entire signal $y[0], \ldots, y[N-1]$:

$$p\left(\mathbf{x}_t|y[0], \ldots, y[N-1], \{\rho_m, \omega_m\}_{m=1,\ldots,M}, \sigma_f^2, \sigma_n^2\right)$$

using the Kalman filtering and smoothing recursions. The only remaining requirement is that a multivariate normal prior $p(\mathbf{x}_0)$ be specified as the initial condition of the state vector.

The unknown parameters for the state-space model appear together as a complex number $a_m \equiv \rho_m \exp[i\omega_m]$. We show that the posterior distribution of the unknown parameters a_m is a normal distribution if a normal

prior

$$p\left(a_m|\mu_m,\sigma_m^2\right)=\mathcal{N}\left(a_m;\mu_m,\sigma_m^2\right) \tag{5.15}$$

is used:

$$p\left(a_m|x_m\left[0\right],\ldots,x_m\left[N-1\right],\mu_m,\sigma_m^2,\sigma_f^2\right)=\frac{1}{Z_x}p\left(x_m\left[0\right],\ldots,x_m\left[N-1\right],a_m,\mu_m,\sigma_m^2,\sigma_f^2\right) \tag{5.16}$$

$$=\frac{1}{Z_x}p\left(a_m,\mu_m,\sigma_m^2\right)\prod_{t=1}^{N-1}p\left(x_m\left[t\right]|x_m\left[t-1\right],a_m,\sigma_f^2\right) \tag{5.17}$$

$$=\frac{1}{Z_x}\exp\left(-\frac{a_m^2}{2\sigma_m^2}+\frac{a_m\mu_m}{\sigma_m^2}+\frac{a_m}{\sigma_f^2}\sum_{t=1}^{N-1}x_m\left[t\right]x_m\left[t-1\right]-\frac{a_m^2}{2\sigma_f^2}\sum_{t=1}^{N-1}x_m^2\left[t-1\right]\right)$$

$$=\frac{1}{Z_x}\exp\left(-\frac{1}{2}\left(\frac{1}{\sigma_m^2}+\frac{1}{\sigma_f^2}\sum_{t=1}^{N-1}x_m^2\left[t-1\right]\right)a_m^2+\left(\frac{\mu_m}{\sigma_m^2}+\frac{1}{\sigma_f^2}\sum_{t=1}^{N-1}x_m\left[t\right]x_m\left[t-1\right]\right)a_m\right) \tag{5.18}$$

where Z_x is the normalizing constant of the posterior distribution of a_m:

$$Z_x=p\left(x_m\left[0\right],\ldots,x_m\left[N-1\right],\mu_m,\sigma_m^2,\sigma_f^2\right)$$

From this we see that the variance of the posterior distribution of a_m is

$$\left(\frac{1}{\sigma_m^2}+\frac{1}{\sigma_f^2}\sum_{t=1}^{N-1}x_m^2\left[t-1\right]\right)^{-1}$$

and the mean is

$$\left(\frac{1}{\sigma_m^2}+\frac{1}{\sigma_f^2}\sum_{t=1}^{N-1}x_m^2\left[t-1\right]\right)^{-1}\left(\frac{\mu_m}{\sigma_m^2}+\frac{1}{\sigma_f^2}\sum_{t=1}^{N-1}x_m\left[t\right]x_m\left[t-1\right]\right)$$

For a known number of partials M we have derived an MCMC scheme to infer the posterior distribution, which is presented as 5.1.

$$p\left(a_1,\ldots,a_M,\mathbf{x}_0,\ldots,\mathbf{x}_{N-1},\sigma_n^2|y\left[0\right],\ldots,y\left[N-1\right],\sigma_f^2,\left\{\mu_m,\sigma_m^2\right\}_{m=1,\ldots,M}\right)$$

Although this is a simple and straightforward MCMC scheme for spectrum estimation, without specifying additional parameters for tuning the algorithm, the requirement that the prior $p\left(a_m\right)$ on the damping ratio and partial frequencies must be a normal distribution is very restrictive. We do not foresee that Bayesian hierarchical models for musical structure such as key and chords will impose normally distributed priors on the partial frequencies. Rather, at this stage, we may allow an arbitrary prior $p\left(a_1,\ldots,a_M\right)$, but the final sampling step of 5.1 must then be replaced with a Metropolis-Hastings step. Our contribution here is to suggest a proposal distribution strongly based on the underlying model of the signal, which has been found in practice to have a high acceptance rate whilst reaching the mode of the posterior distribution of the damping ratio and partial frequencies rapidly. This contrasts with global proposals based on the periodogram estimate and local random-walk proposals normally required to effectively explore the non-linear posterior (see 5.3.4).

Algorithm 5.1 Gibbs sampler for the state-space model

- Initialization
  - For $m = 1, \ldots, M$ sample the diagonal elements of $\mathbf{A}^{(0)}$: $a_m^{(0)} \sim p\left(a_m|\mu_m, \sigma_m^2\right)$(5.15))
  - Sample $\mathbf{x}_0^{(0)} \sim p\left(\mathbf{x}_0\right)$
  - For $t = 1, \ldots, N-1$ sample $\mathbf{x}_t^{(0)} \sim p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(0)}, \mathbf{A}^{(0)}, \sigma_f^2\right)$(5.14)
- Iterations, $i = 1, 2, \ldots$
  - Compute $n^{(i-1)}\left[t\right] = y[t] - \sum_{m=1}^{m} x_m^{(i-1)}\left[t\right]$ and sample $\sigma_n^{2(i)} \sim p\left(\sigma_n^2|\mathbf{n}^{(i-1)}\right)$(5.12)
  - For $t = 1, \ldots, N-1$ sample $\mathbf{x}_t^{(i)} \sim p\left(\mathbf{x}_t|y\left[0\right], \ldots, y\left[N-1\right], \mathbf{A}^{(i-1)}, \sigma_f^2, \sigma_n^{2(i)}\right)$ computed using Kalman filter and smoother recursions
  - For $m = 1, \ldots, M$ sample $a_m^{(i)} \sim p\left(a_m|x_m^{(i)}\left[0\right], \ldots, x_m^{(i)}\left[N-1\right], \mu_m, \sigma_m^2, \sigma_f^2\right)$ (5.18)

In this scheme, for each $m = 1, \ldots, M$ we use a proposal distribution

$$Q\left(a_m; x_m^{(i)}\left[0\right], \ldots, x_m^{(i)}\left[N-1\right], \sigma_f^2, a_{1:m-1}^{(i)}, a_{m+1:M}^{(i-1)}\right)$$

which is of the same form as (5.18) but substituting

$$\mu_m = \left\langle a_m \right\rangle_{p\left(a_m|a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)}$$

and

$$\sigma_m^2 = \left\langle a_m^2 \right\rangle_{p\left(a_m|a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)} - \mu_m^2$$

so that the proposal distribution would be equal to the posterior if the prior were a normal distribution. The acceptance probability of the proposed candidate $a_m' \sim Q\left(a_m; x_m^{(i)}\left[0\right], \ldots, x_m^{(i)}\left[N-1\right], \sigma_f^2, a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)$ is the minimum of 1 and

$$\frac{\prod_{t=1}^{N-1} \mathcal{N}\left(x_m^{(i-1)}\left[t\right]; a_m' x_m^{(i-1)}\left[t-1\right], \sigma_f^2\right) p\left(a_m'|a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)}{\prod_{t=1}^{N-1} \mathcal{N}\left(x_m^{(i-1)}\left[t\right]; a_m^{(i-1)} x_m^{(i-1)}\left[t-1\right], \sigma_f^2\right) p\left(a_m^{(i-1)}|a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)} \times \frac{Q\left(a_m^{(i-1)}; x_m^{(i)}\left[0\right], \ldots, x_m^{(i)}\left[N-1\right], \sigma_f^2, a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)}{Q\left(a_m'; x_m^{(i)}\left[0\right], \ldots, x_m^{(i)}\left[N-1\right], \sigma_f^2, a_{1:m-1}^{(i-1)}, a_{m+1:M}^{(i)}\right)}$$

5.3.4 Gabor Model

The Gabor formulation of the model is given by (5.8)

$$y\left[t\right] = \sum_{m=1}^{M} \sum_{i=0}^{I} b_{i,m} \psi\left[t - i\Delta\right] \exp\left[i\omega_m t\right] + n\left[t\right] \tag{5.19}$$

For convenience, we rewrite (5.19) in matrix form, by stacking the amplitudes $b_{i,m}$ into a column vector $\mathbf{b}$ of length $(I+1)M$, with elements

$$\mathbf{b}_{(m-1)(I+1)+i} = b_{i,m}$$

and the Gabor basis functions into a $N \times (I+1)M$ matrix $\mathbf{D}$ with elements

$$\mathbf{D}_{t,(m-1)(I+1)+i} = \psi[t - i\Delta] \exp[i\omega_m t] \tag{5.20}$$

Writing $y = [y[0], \ldots, y[N-1]]^\top$ and $\mathbf{n} \equiv [n[0], \ldots, n[N-1]]^\top$ as before, (5.19) becomes

$$y = \mathbf{D}\mathbf{b} + \mathbf{n}$$

The approaches described in 5.3.1 related to this model all adopt the g-prior [Zellner, 1986] which is chosen for its properties in Bayesian model selection. The g-prior is a zero mean multivariate normal prior distribution for $\mathrm{p}\left(\mathbf{b}|\mathbf{D}, \sigma_n^2\right)$ with covariance matrix $\sigma_n^2 \xi \left(\mathbf{D}^\top \mathbf{D}\right)^{-1}$ and an additional parameter ξ. As ξ scales the amplitudes with respect to the noise level, it can be interpreted as a prior signal-to-noise ratio. In the case where we treat ξ as unknown and wish to additionally infer it in a Bayesian setting, we again follow the literature in 5.3.1[1] and assign an inverse-gamma prior: $p(\xi) = \mathcal{IG}(\xi; \alpha_\xi, \beta_\xi)$.

The probabilistic model described so far, including the additional parameter ξ which was introduced by adopting the g-prior, is

$$p\left(y, \mathbf{b}, \mathbf{D}, \sigma_n^2, \xi\right) = p\left(y|\mathbf{D}\mathbf{b}, \sigma_n^2\right) p\left(\mathbf{b}|\mathbf{D}, \sigma_n^2, \xi\right) p\left(\sigma_n^2\right) p(\xi) \tag{5.21}$$

We have yet to discuss a prior $p(\mathbf{D})$ for $\mathbf{D}$. From (5.20) this is a prior $p(\omega_1, \ldots, \omega_M)$ on the partial frequencies, which are the remaining unknowns in this model.

In the remainder of this section, we show a result for this model that has not been referred to in the literature we have reviewed. The model parameters $\mathbf{b}$ and σ_n^2 may be integrated out, giving the following marginal distribution:

$$p(y|\mathbf{D}, \xi)\, p(\xi) \propto \left(y^\top \mathbf{P} y + \beta_n\right)^{-(N+\alpha_n)/2} \xi^{-(\alpha_\xi+1)} \exp\left(-\frac{\beta_\xi}{\xi}\right) \tag{5.22}$$

where the following definitions are used, as in the literature:

$$\begin{aligned}
\mathbf{S}^{-1} &= \mathbf{D}^\top \mathbf{D} + \frac{1}{\xi}\mathbf{D}^\top \mathbf{D} \\
&= \frac{\xi+1}{\xi}\mathbf{D}^\top \mathbf{D} \\
\mathbf{P} &= \mathbf{I}_N - \mathbf{D}\mathbf{S}\mathbf{D}^\top \\
&= \mathbf{I}_N - \frac{\xi}{\xi+1}\mathbf{D}\left(\mathbf{D}^\top \mathbf{D}\right)^{-1}\mathbf{D}^\top \\
&= \mathbf{I}_N - \frac{\xi}{\xi+1}\mathbf{D}\mathbf{D}^\dagger
\end{aligned} \tag{5.23}$$

[1] For clarity of notation we denote the scaling hyperparameter as ξ. This is equivalent to δ^2 used in Andrieu and Doucet [1999] and ξ^2 used in Davy et al. [2006]

where $\mathbf{D}^\dagger = \left(\mathbf{D}^\top\mathbf{D}\right)^{-1}\mathbf{D}^\top$ has been used to make the notation a little more concise. It would be useful for Bayesian inference to be able to sample from the conditional distribution $p\left(\xi|y,\mathbf{D}\right)$ but this is not a standard distribution. Rather instead, previous approaches have used the following distribution

$$p\left(\xi|\mathbf{D},\mathbf{b},\sigma_n^2\right) = \mathcal{IG}\left(\alpha_\xi + (I+1)\,M, \frac{1}{2\sigma_n^2}\mathbf{b}^\top\mathbf{D}^\top\mathbf{D}\mathbf{b} + \beta_\xi\right)$$

which is a standard distribution, but requires that $\mathbf{b}$ and σ_n^2 be available. This may not be satisfactory given we chose to integrate them out analytically in (5.22) thus improving the Monte-Carlo estimates of the remaining parameters because significant additional computation is required to simulate them. Alternatively, it is possible to integrate $p\left(\xi|y,\mathbf{D}\right)$ numerically, as Richardson and Green [1997] have chosen to do.

The result we present here is that although $p\left(\xi|y,\mathbf{D}\right)$ is not a standard distribution, its mode, or MAP estimate, is available as the solution of the quadratic equation (see Section B.1 for the full derivation)

$$\xi^2\left(\frac{N+\alpha_n}{2} + (\alpha_\xi+1)\right)\left(y^\top\mathbf{D}\mathbf{D}^\dagger y\right) - \xi\left((\alpha_\xi+1)\,y^\top y + \beta_\xi y^\top\mathbf{D}\mathbf{D}^\dagger y\right) + \beta_\xi y^\top y \quad = \quad 0 \qquad (5.24)$$

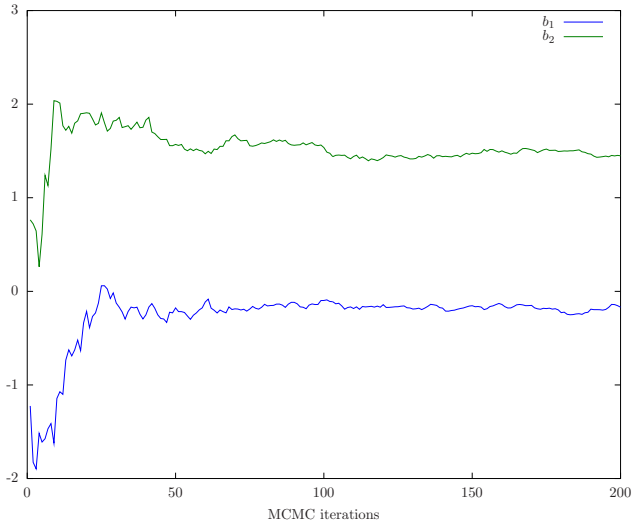
The positive root of (5.24) is the mode, the other root is negative and is disallowed by the prior $p\left(\xi\right)$.

Using (5.24) to estimate the mode ξ^* reduces the computation required for each iteration of the MCMC algorithm, and slightly reducing the number of iterations required for convergence. This is illustrated in Figure 5.2 on page 68 for a single violin note with fundamental frequency 440Hz, which is part of the data set used later in this chapter in 5.5.1 to demonstrate the ability of these algorithms to infer partial frequencies in the signal.

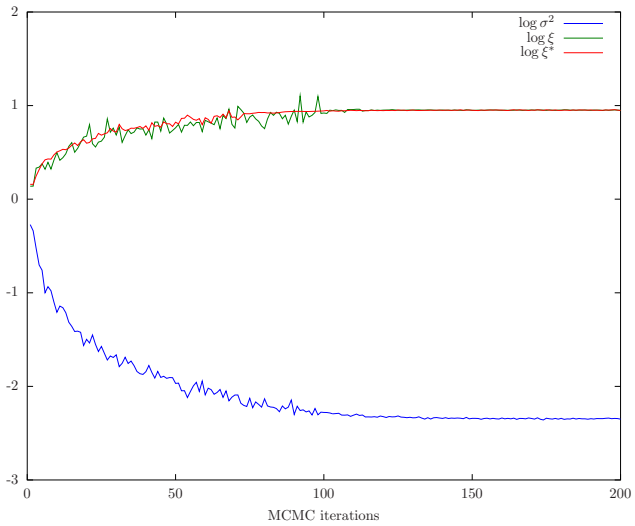
This result is useful in an iterative MAP estimation scheme. It can also be used as the basis of a proposal distribution for a Metropolis Hastings MCMC kernel. As the prior $p\left(\xi\right)$ and conditional posterior distribution $p\left(\xi|\mathbf{D},\mathbf{b},\sigma_n^2\right)$ are both inverse gamma, it seems plausible to construct a proposal distribution which is also inverse gamma, with its mode ξ^* given by (5.24). We suggest setting the shape parameter of the proposal distribution to α_q and the scale parameter to $(\alpha_q+1)\,\xi^*$. α_q controls the variance of the proposal and should be tuned for a suitable acceptance rate.

A Metropolis Hastings kernel is also necessary for simulating the frequencies $\omega_1,\ldots,\omega_M$ as the joint distribution $p\left(\omega_1,\ldots,\omega_M|\xi,y\right)$ or any of the individual distributions $p\left(\omega_m|\omega_1,\ldots,\omega_{m-1},\omega_{m+1},\ldots,\omega_M,\xi,y\right)$ are not standard distributions. We are able to impose any prior distribution $p\left(\omega_{1:M}\right)$ using the Gabor model. Here we suggest two proposal distributions based on computing the residual of the signal excluding the partial frequency of interest. Denote $\mathbf{D}_{\neg m}$ as the $N\times(I+1)\,(M-1)$ matrix where the columns related to the mth partial have been omitted. The least squares reconstruction of the signal excluding the mth partial is given by $\mathbf{D}_{\neg m}\mathbf{D}_{\neg m}^\dagger y$, and the residual signal, which is expected to contain the mth partial and also noise, is given by $\mathbf{x}_m = y - \mathbf{D}_{\neg m}\mathbf{D}_{\neg m}^\dagger y$. Note that computing the residual avoids simulating $\mathbf{b}$ and σ_n^2. The proposal distributions are thus of the form $Q\left(\omega_m; y, \omega_{1:M\neg m}, \xi\right)$ where $\omega_{1:M\neg m}$ denotes the set of partial frequencies excluding ω_m. We are taking advantage of being able to design custom Metropolis Hasting kernels in order to reduce computation whilst performing full Bayesian inference.

The first proposal involves computing a K-point DFT of $\mathbf{x}_m$ and defining a probability distribution function proportional to the magnitude in each frequency bin. This allows the algorithm to explore many parts of the spectrum rapidly. For the proposal to be reversible, the proposed frequency is assumed to be



(a) The amplitudes of the Gabor basis functions must be simulated in order to sample from the posterior of the signal-to-noise ratio parameter. This figure presents the convergence of two amplitude parameters of the fundamental frequency. The Markov chain has converged to the true posterior distribution in approximately 100 iterations. The convergence of the corresponding noise variance and signal-to-noise ratio parameters for this Markov chain is shown in Figure 5.2b.



(b) Convergence of the noise variance σ^2 and a comparison of the convergence of ξ when inferred from the posterior distribution, or set to the mode ξ^*. As in Figure 5.2a, the simulated parameters converge in approximately 100 iterations, whereas the convergence to the mode is complete in around 75 iterations.

Figure 5.2: The result presented in (5.24) allows the conditional mode of the signal-to-noise parameter ξ to be calculated without requiring inference of the amplitude parameters $\mathbf{b}$ or the noise variance σ^2, which reduces the computation required for each iteration of the MCMC algorithm, and slightly reducing the number of iterations required for convergence.

uniformly distributed within the range of frequencies $|p/K - \omega_m| < 1/2$ for the frequency bin p with centre frequency p/K, i.e.,

$$Q\left(\omega_m; y, \omega_{1:M\neg m}, \xi\right) \propto \prod_{p=0}^{K-1} \begin{cases} \left|DFT\left[\mathbf{x}_m\right]\right|_p & \left|\frac{p}{K} - \omega_m\right| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.25}$$

The second proposal is based on fitting a damped sinusoid to the residual signal, discarding the damping ratio and using the frequency obtained. This proposal is used to make small adjustments to the partial frequencies to search for local maxima in the posterior distribution. We use the posterior distribution (5.18) which we derived for a damped sinusoid with process noise. In this MCMC context, we can use σ_f^2, the process noise variance, to control the acceptance rate, and the damping ratio ρ_m is discarded. For simplicity, we also remove the prior parameters μ_m, σ_m^2, giving

$$Q\left(\omega_m; y, \omega_{1:M\neg m}, \xi\right) = \mathcal{N}\left(\omega_m; \left(\sum_{t=1}^{N-1} x_m^2\left[t-1\right]\right)^{-1}\left(\sum_{t=1}^{N-1} x_m\left[t\right] x_m\left[t-1\right]\right), \sigma_f^2\left(\sum_{t=1}^{N-1} x_m^2\left[t-1\right]\right)^{-1}\right) \tag{5.26}$$

Alternatively we may use a standard random-walk proposal to search for a local maxima, especially in cases where the above damped sinusoid model is not appropriate, such as for sustained note instruments with significant amplitude and frequency modulations. The proposal distribution is a Gaussian distribution centered around the existing frequency with a small random-walk variance σ_{RW}^2 which Godsill and Davy [2005] suggest setting to 10^{-3}.

$$Q\left(\omega; \omega_m\right) = \mathcal{N}\left(\omega; \omega_m, \sigma_{\mathrm{RW}}^2\right) \tag{5.27}$$

5.4 Bayesian Inference using Reversible Jump MCMC

In the previous section we have developed two probabilistic signal models for musical signals. The state-space model is considered useful for musical instruments which may be approximately modelled by damped oscillators. The Gabor basis model is considered useful for bandlimited frequency and amplitude modulations such as those arising from vibrato.

MCMC algorithms (5.1 and 5.2) have been derived for a fixed number M of partial frequencies with conditionally independent priors $p\left(\omega_{1:M}|M\right)$. However there are few situations where the number of partials and their frequencies are known *a priori*. In this section we consider how to infer the number of partials jointly with their frequencies. The most flexible method of Bayesian inference for this type of problem is reversible jump MCMC, which was first applied to Bayesian sinusoidal models in Andrieu and Doucet [1999].

Firstly we must specify a prior $p\left(M\right)$ on the overall number of partials. In the literature the prior on the number of partials is typically split into a hierarchical model, with a prior on the number of notes and a prior on the number of partials in each note.

The contribution here is our general presentation of how to propose and accept changes to the numbers of partials and their frequencies, without referring to any specific move, and also applying reversible jump MCMC to the state-space model. Being able to propose changes to multiple partials is necessary for being able to rapidly explore the numerous combinations of harmonics and notes that arise in musical signals.

Algorithm 5.2 Metropolis-Hastings for the Gabor model

- Initialization
  - For $m = 1, \ldots, M$ sample $\omega_m^{(0)} \sim p(\omega_m)$
  - Sample $\xi^{(0)} \sim \mathcal{IG}(\alpha_\xi, \beta_\xi)$
- Iterations $i = 1, 2, \ldots$
  - Choose m from $1, \ldots, M$ with equal probability $1/M$
  - Compute $\mathbf{x}_m = y - \mathbf{D}_{\neg m}^{(i-1)} \mathbf{D}_{\neg m}^{\dagger(i-1)} y$
  - Select a proposal $Q\left(\omega_m; y, \omega_{1:M}^{(i-1)}, \xi^{(i-1)}\right)$ to sample ω_m' from ((5.25)–(5.27))
  - draw u from $\mathcal{U}(0,1)$ and set $\omega_m^{(i)} \leftarrow \omega_m'$ if

$$u < \frac{p\left(y, \mathbf{D}', \xi^{(i-1)}\right)}{p\left(y, \mathbf{D}^{(i-1)}, \xi^{(i-1)}\right)} \frac{Q\left(\omega_m^{(i-1)}; y, \omega_{1:M}^{(i-1)}, \xi^{(i-1)}\right)}{Q\left(\omega_m'; y, \omega_{1:M}^{(i-1)}, \xi^{(i-1)}\right)}$$

  - otherwise set $\omega_m^{(i)} \leftarrow \omega_m^{(i-1)}$
  - Compute ξ^* using (5.24)
  - Sample ξ' from $Q\left(\xi; \mathbf{D}^{(i)}, y\right) \equiv \mathcal{IG}\left(\xi; \alpha_q, (\alpha_q + 1)\xi^*\right)$
  - draw v from $\mathcal{U}(0,1)$ and set $\xi^{(i)} \leftarrow \xi^{(i-1)}$ if

$$v < \frac{p\left(y|\mathbf{D}^{(i)}, \xi'\right) p(\xi')}{p\left(y|\mathbf{D}^{(i)}, \xi^{(i-1)}\right) p\left(\xi^{(i-1)}\right)} \frac{Q\left(\xi^{(i-1)}; \mathbf{D}^{(i)}, y\right)}{Q\left(\xi'; \mathbf{D}^{(i)}, y\right)}$$

5.4.1 Proposals and Acceptance Ratios

In the following expressions in this section, $\omega_{1:M}$ should be substituted with $a_{1:M}$ when using the state-space model. When using the Gabor model, all of the joint and proposal distributions are dependent on the current value of $\xi^{(i)}$.

A proposal distribution for changing the number of partials and their frequencies would be expressed as follows

$$Q\left(\omega_{1:M}, M|y, \omega_{1:M}^{(i-1)}, M^{(i-1)}\right)$$

however this proposal distribution is not reversible as the dimension of the model parameters has changed.

Instead we define a birth proposal, where the number of partials has increased, but the frequencies of the original partials have not changed, i.e., $M^* > M^{(i-1)}$

$$\left\{\omega^*_{\left(M^{(i-1)}+1\right):M^*}, M^*\right\} \sim Q_B\left(\omega_{\left(M^{(i-1)}+1\right):M}, M|y, \omega_{1:M^{(i-1)}}^{(i-1)}, M^{(i-1)}\right)$$

such that if the proposal is accepted, $M^* \to M^{(i)}$ and $\left\{\omega_{1:M^{(i-1)}}^{(i-1)}, \omega^*_{\left(M^{(i-1)}+1\right):M^*}\right\} \to \omega_{1:M^{(i)}}^{(i)}$.

This proposal must be accompanied by a death proposal, where a number of the original partials have been removed, and the remaining partial frequencies unchanged, i.e., $M^* < M^{(i-1)}$

$$M^* \sim Q_D\left(M|y, \omega_{1:M^{(i-1)}}^{(i-1)}, M^{(i-1)}\right)$$

such that if the proposal is accepted, $M^* \to M^{(i)}$ and $\omega_{1:M^{(i-1)}}^{(i-1)} \to \omega_{1:M^{(i)}}^{(i)}$.

The Jacobian of the transform for the birth and death proposals equals one. The acceptance ratio for a birth proposal is given by the minimum of 1 and

$$\frac{p\left(y, \omega_{1:M}^{(i-1)}, \omega^*_{\left(M^{(i-1)}+1\right):M^*}, M^*\right)}{p\left(y, \omega_{1:M^{(i-1)}}^{(i-1)}, M\right)} \frac{Q_D\left(M^{(i-1)}|y, \omega_{1:M^{(i-1)}}^{(i-1)}, \omega^*_{\left(M^{(i-1)}+1\right):M^*}, M^*\right)}{Q_B\left(\omega^*_{\left(M^{(i-1)}+1\right):M^*}, M^*|y, \omega_{1:M^{(i-1)}}^{(i-1)}, M^{(i-1)}\right)}$$

Similarly, the acceptance ratio for a death proposal is given by the minimum of 1 and the reciprocal of the above ratio:

$$\frac{p\left(y, \omega_{1:M^*}^{(i-1)}, M^*\right)}{p\left(y, \omega_{1:M^{(i-1)}}^{(i-1)}, M^{(i-1)}\right)} \frac{Q_B\left(\omega_{(M^*+1):M^{(i-1)}}, M^{(i-1)}|y, \omega_{1:M^*}^{(i-1)}, M^*\right)}{Q_D\left(M^*|y, \omega_{1:M^{(i-1)}}^{(i-1)}, M^{(i-1)}\right)}$$

Before or after any MCMC move, a complete reordering of the partial frequencies $\omega_{1:M}$ is permitted, so that different combinations of partials may be created or destroyed.

5.4.2 Prior Model

For the remainder of this chapter we adopt the hierarchical prior used by Godsill and Davy [2005] for the partial frequency structure of the music. A short segment of music has K notes, where the unknown number

of notes is distributed according to a truncated Poisson distribution:

$$p\left(K=k|\lambda_K\right)=\frac{\left(\lambda_K^k/k!\right)}{\sum_{k'=K_{\mathrm{MIN}}}^{K_{\mathrm{MAX}}}\left(\lambda_K^{k'}/k'!\right)} \tag{5.28}$$

The hyperparameter λ_K is distributed according to a Gamma distribution $p\left(\lambda_K\right)=\mathcal{G}\left(\lambda_K;\alpha_K,\beta_K\right)$ where $\alpha_K=1$ and $\beta_K=2$ are set such that $p\left(\lambda_K\right)$ has infinite variance. It is often reasonable within the context of polyphonic music transcription to assume that the minimum and maximum number of notes sounding at the same time is known in advance (for example a four-part fugue or choral).

Each note $k=1,\ldots.K$ has M_k partials which is distributed according to the same truncated Poisson distribution (5.28). The minimum number of partials is set to 2 in the prior work, and the maximum number of partials is set to either 30 or the limit and $\omega_s/2\omega_0$ which is the number of partials permitted by the sampling frequency. Here we have set the minimum number of partials to 1 and allowed as many partials as the sampling frequency permits. The sampling of the hyperparameters for the posterior distributions of the number of notes and partials is identical to that of Andrieu and Doucet [1999].

The prior distribution for frequencies $p\left(\omega_{1:M_k}|M_k\right)$ is given by the following hierarchical model

$$\begin{aligned} p\left(\omega_{1:M_k}|M_k\right) &= p\left(\omega_0\right)\prod_{k=1}^{M_k}p\left(\delta_k\right) \\ \omega_k &= k\omega_0\left(1+\delta_k\right) \\ p\left(\delta_k\right) &= \mathcal{N}\left(0,\sigma_\delta^2\right) \end{aligned}$$

with $\sigma_\delta^2=3\times10^{-8}$. $p\left(\omega_0\right)$ is set to be uniform, with limits set to be a semitone below and above the minimum and maximum MIDI frequencies.

5.4.3 Examples of Reversible Moves

The following moves, designed for inferring the harmonic structure of polyphonic music, are taken from Godsill and Davy [2005] for the prior model described in the previous section.

5.4.3.1 n-increase/decrease

This pair of moves is designed to estimate the number of harmonics present in the signal for a single note. For the n-increase move, a new set of n partial frequencies are proposed in harmonic positions above the highest existing harmonic for that note. For the n-decrease move, the highest n partial frequencies for that note are deleted. The number of possible harmonics is limited between 1 and $\omega_s/2\omega_0$ where ω_s is the sampling frequency and ω_0 is the fundamental frequency of the note.

5.4.3.2 double/halve frequency

This pair of moves is designed to explore octave ambiguities and errors in the pattern of harmonics in the signal. The *double* move doubles the fundamental frequency of a note by removing the odd-numbered partials of that note. The *halve* move halves the fundamental frequency of a note, keeping the existing partial

frequencies, but assigning them to twice the original harmonic position. The missing partials are added in the odd-number harmonic positions.

5.4.3.3 note birth/death

This pair of moves is designed to estimate the number of notes in the signal. The birth move adds a new note, with a fundamental frequency and a new set of harmonics. The death move deletes a note with all of its partial frequencies.

5.5 Results

In this chapter we have extended and developed two existing models for musical signal analysis, and described a reversible jump MCMC framework to infer the number of partials in a small segment of music, and their respective partial frequencies. To put this work into perspective, in this section we use the Bayesian prior model of Davy et al. [2006] for polyphony and harmonic notes to infer the number of notes and their fundamental frequencies in short segments of musical chords.

The objective of this section is to quantitatively evaluate the effect of the model enhancements suggested in this chapter on polyphonic music transcription, using the prior work as a baseline. In Davy et al. [2006], multiple F0 estimation results are presented for a set of recorded note mixtures of a variety of musical instruments from the McGill database[2]. Each signal is downsampled from 44,100 Hz to 11,025 Hz, and the first 544ms is used to estimate the fundamental frequencies present, assuming the number of notes is known *a priori*. We use the same experimental setup, running the reversible jump MCMC algorithm once for 800 iterations, and using only the final 100 samples to estimate the fundamental frequencies and the frequencies of the partials.

5.5.1 Performance on Monophonic Extracts

When evaluating model-based polyphonic music transcription systems, it is implicitly expected that the system should accurately detect the fundamental frequency of an isolated note in a monophonic extract, and be resilient to octave errors, as this capability already exists in simpler detection systems based on autocorrelation function or harmonic transform analysis. We have found it extremely useful to study how the partial frequencies are detected by a model, and have found that significant differences exists in the performance of the different models suggested in this chapter. We choose to study monophonic extracts, as the actual number of partials existing in the signal can be estimated easily by hand by studying the periodogram of the entire monophonic extract. If a model is able to successfully detect a higher number of partials, this indicates that the model is able to capture more of the harmonic structure of a note, and should therefore be able distinguish better between ambiguous combinations of notes that degrade the performance of polyphonic music transcription systems.

In this section, we present partial estimation results for different model choices. In each case, we have modified one aspect of the model, and ensured that all other model and algorithm parameters are constant. A ground truth for the number of partials was prepared by a human observer from the periodogram with

[2] `http://www.music.mcgill.ca/resources/mums/html/MUMS_dvd.htm`

		Average percentage of partials detected	
Instrument	Number of Extracts	Real Signal	Analytical Representation
Violin	31	37.3	73.3
French Horn	36	50.5	82.5
Oboe	40	48.2	80.8
Flute	29	47.5	77.5
Trumpet	26	51.0	81.0
Clarinet	34	47.7	78.5
Viola	32	40.0	66.7
Piano	64	47.2	74.1
Guitar	48	44.7	76.6
TOTAL	340	46.1	76.6

Table 5.1: Partial estimation results, comparing the average percentage of partials detected from the real signal and the analytical representation using the Gabor model with Hamming basis functions

prior knowledge of the instrument and the fundamental frequency of the note. For each extract, we record the number of partials detected by the model compared to the ground truth number of partials, and express the ratio as a percentage. The partial estimation results are grouped by musical instrument, and the percentage of partials detected is averaged over the extracts for that instrument[3]. The results are presented in Table 5.1 on page 74 and Table 5.2 on page 75 for a variety of instruments over a range of pitches from the McGill database.

The first set of results compares choosing to apply the model to the original real signal or to the Hilbert transform of the signal. In Table 5.1 on page 74 we use the Gabor model with Hamming window basis functions, and compare the number of partials modelled using the original signal and using its Hilbert transform. It is clear from these results that applying the Hilbert transform, which reduces some of the modelling ambiguity in the sinusoidal representation of the signal, increases the number of partials correctly estimated from the note.

The second comparison we make is between the choice of basis function, or the use of the state space model, when applied to the analytical representation of the signal. Partial estimation results are presented in Table 5.2 on page 75 comparing the number of partials modelled using the Hamming, Gaussian and sinc basis functions and the state-space model. The difference in the number of partials correctly estimated is overall much less dramatic. The Gaussian basis and the state-space model perform slightly worse than the Hamming and Sinc bases, which themselves mostly return a consistent number of partials. However, as motivated in Section 5.2, the performance for certain types of instruments can be improved by the choice of model. For violin and viola notes, which are played with substantial vibrato, the sinc basis model detects more partials than the other models. The higher order partials hence have a high spread in spectral energy in frequency, and it appears that only the sinc basis model is correctly detecting and modelling these frequency modulations. This can be explained by the explicit bandwidth constraint on frequency and amplitude modulations when using a sinc basis. For piano and guitar notes, which are played percussively and allowed to decay over time, the state-space model, with a constant damping ratio across the length of note, detects

[3]As an illustration, if we have two oboe extracts, the first having 10 partials and 6 were detected by the model (60%), and the second having 12 partials and 9 detected by the model (75%) then we record the average percentage of partials detected for this model and instrument as 67.5%

		Average percentage of partials detected			
Instrument	Number of Extracts	Hamming	Sinc	Gaussian ($\alpha = 2.5$)	State-space
Violin	31	73.3	84.4	59.3	59.8
French Horn	36	82.5	83.9	69.2	70.2
Oboe	40	80.8	74.7	69.6	68.1
Flute	29	77.5	79.9	67.9	70.9
Trumpet	26	81.0	76.5	70.3	60.8
Clarinet	34	78.5	80.0	68.4	71.9
Viola	32	66.7	78.3	61.7	61.3
Piano	64	74.1	73.2	70.3	90.7
Guitar	48	76.6	75.4	69.2	90.4
TOTAL	340	76.6	77.8	67.7	74.4

Table 5.2: Partial estimation results for different instruments, comparing Hamming, sinc and Gaussian basis functions and the state-space model with constant damping ratio, on the analytical representation of the signal

nearly all of the partials of these notes.

To conclude, we have found that using the Hilbert transform to calculate the analytical representation of a signal, and using this representation for modelling, increases the number of partials that are detected and modelled correctly in single harmonic notes. The additional computation required to compute the transform is small, and we expect this to increase polyphonic music transcription by capturing more higher order structure in harmonic notes.

We have also shown that in most cases that there is little difference in the partial estimation performance of different basis functions, or using the state-space model. Therefore, we suggest that, unless there is appreciable vibrato or other modulations, Hamming basis functions should be used as they have finite support and therefore it is more efficient to compute and invert the basis. In cases where the modulation bandwidth is known a priori and is sufficiently large, sinc basis functions are attractive as we have observed that higher order partials are modelled correctly, whereas other basis functions will typically either split a higher order partial into two adjacent frequencies, or fail to detect the frequency. For instruments with an approximately constant damping ratio over the length of the note, we have shown that the state-space model defined in this chapter is appropriate for modelling the higher order partials of the signal.

5.5.2 Multiple F0 Estimation

In this section, we present results for polyphonic transcription where the number of notes are known *a priori*. This is a limited application, as this situation is rare in practice. Similar implementations of Bayesian inference for these harmonic models overestimate the number of notes, and have to resort to a somewhat unsatisfactory heuristic to determine whether the energy of a modelled note is too low to be significant. In our research with these models, we have experienced the same problem, however we have also determined that the difficulty does not lie with the model, but the estimation accuracy of the partial frequencies. We have made the following observations:

1. The local maxima of the likelihood of a single partial frequency also minimizes the residual signal in the vicinity of that frequency. Correctly minimizing the residual of the signal around a partial frequency

		Notes in Mixture			
Evaluation Metric	Model	1	2	3	4
% octave error	Gabor	0	2.8	11.1	10.2
	Davy et al. [2006]	0	10.3	17.8	9.3
	Klapuri [2008]	0	13.6	19.0	22.2
% pitch error	Gabor	0	8.3	15.6	18.6
	Davy et al. [2006]	0	5.1	7.2	19.7
	Klapuri [2008]	0	1.4	6.0	10.3
% total error	Gabor	0	11.1	26.7	28.8
	Davy et al. [2006]	0	15.4	25.0	29.0
	Klapuri [2008]	0	15.0	25.0	32.5

Table 5.3: Polyphonic pitch estimation using the Bayesian harmonic model. The total error for the system presented in this chapter is split into errors when the pitch estimated is an octave above or below the ground truth, and errors when the estimated pitch is not an octave error.

reducing the chance of spurious partial detections to either side of the original frequency, which leads to duplicate notes being estimated.

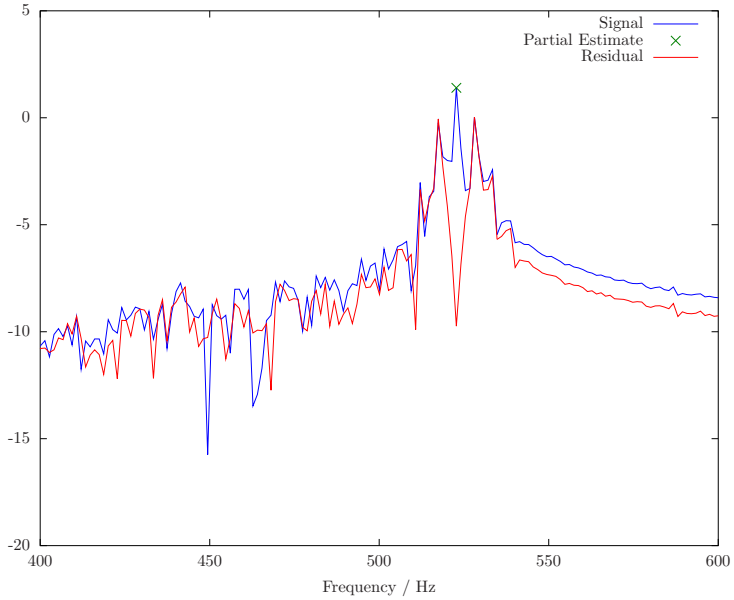
2. Due to frequency and amplitude modulations in the partial, the maximum likelihood frequency may differ by up to 3Hz from the ideal harmonic frequency for theoretically harmonic instruments, and by a comparable amount from the local maxima of the periodogram estimator. This means that global frequency proposals based on these may produce inaccurate initial frequency estimates.

3. The difference between an initial frequency estimate and the maximum likelihood frequency is too large for the constant variance random walk MH proposal distribution used in our algorithm to both arrive near the optimum and accurately estimate its value.

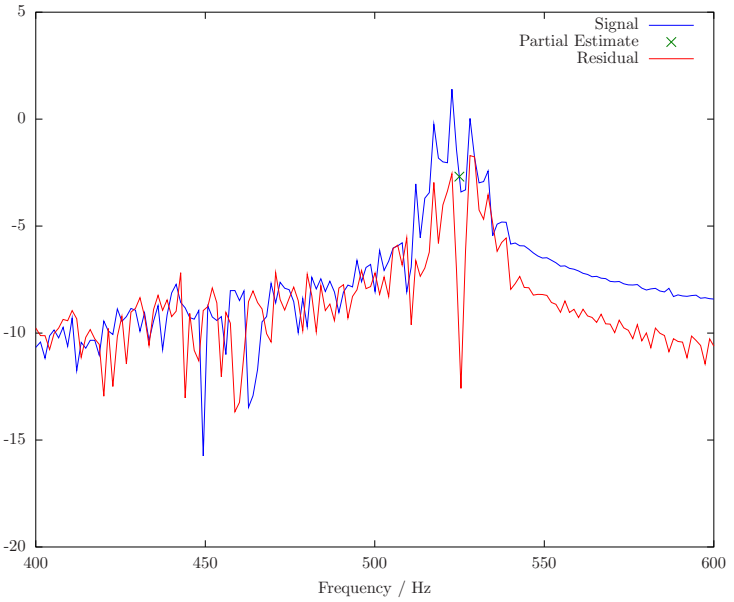
We illustrate these observations in Figure 5.3 on page 77 for a single partial frequency estimated firstly from the local maxima of the periodogram, and then by minimizing the residual of the signal.

We obtained some improvements by adapting the random walk MH proposal to progressively reduce its variance, so that larger jumps in frequency were followed by smaller steps to explore the local maxima. It was also necessary to reduce the ratio of global to local MH proposals so that enough time was given to search for the maximum before losing the current frequency found in a global jump. However this also increased the computation time, and also restricted the algorithms ability to explore the entire spectrum using global proposals. In the next chapter, we show the benefits of estimating the partial frequencies to high accuracy, and how it can be used to effectively estimate the number of notes. As future lines of research using these Bayesian harmonic models with reversible-jump MCMC, we suggest either using a Hamiltonian Monte Carlo scheme to use derivative information in the likelihood to arrive at the local maxima quicker, or use the numerical methods described in Chapter 6 as the basis of a proposal distribution, and balance this correctly with global proposals.

To complete this chapter we present the estimation results where the number of notes are known, so that they can be compared with other transcription systems using this metric. To compare objectively against prior work, we use the data set of Davy et al. [2006] which consists of 20 short monophonic signals and 20 short polyphonic signals for each of 2, 3 and 4 note mixtures, taken from a limited set of instruments



(a) Estimate of the partial frequency from the maxima of the periodogram. The residual of the resulting signal has two prominent peaks on either side of the detected partial frequency. Subsequent iterations of the algorithm will identify these peaks as additional partials, whereas these peaks are in reality due to amplitude and frequency modulations in the original partial.



(b) Maximum likelihood estimation of a partial frequency using the signal model. This estimate minimizes the residual of the signal. The deviation between the peak of the periodogram and the maximum likelihood estimate is 2.2Hz.

Figure 5.3: Comparison of the residual signal when using a periodogram estimate and maximum likelihood estimate of the frequency of a partial of a musical note. The estimated frequency is marked on the periodogram estimate of the signal.

from the McGill database. As this set of notes are only for sustained note instruments and do not include piano or guitar notes, we only use the Gabor model with hamming basis functions for our comparison. Additionally we have prepared an implementation of the auditory model based system of Klapuri [2008] as a state-of-the-art polyphonic transcription to compare against. We present our results in Table 5.3 on page 76. The results show an improvement in transcription accuracy for two note mixtures when using the analytical representation, and we suggest that this is due to the improved estimation of higher partial frequency structure demonstrated in the previous section. However, for three and four note mixtures, the performance is not appreciably different to prior work of Davy et al. [2006], although there is an improvement in accuracy over the auditory model system for 4 note mixtures. We suggest that as the inference algorithm is mostly identical, the inference algorithm is limiting the performance in these cases. During this work, we took the opportunity to study the reversible jump MCMC algorithm in progress, observing the current state of the fitted model after each iteration. We observed that in many of the situations where transcription errors occurred, the algorithm did reach the correct configuration of notes at some point, but was not able to sustain this state due to inaccuracies in the frequency estimates.

5.6 Conclusion

In this chapter we have described and developed signal models for pitched, harmonic instruments from first principles. Our primary motivation has been to investigate the modelling of different forms of frequency and amplitude variations throughout the length of a musical note with a nominally constant frequency. A damped amplitude envelope, where the oscillations decay exponentially in time, was found to fit naturally in a state-space formulation, appropriate for percussive instruments such as the piano and guitar. A limit on the bandwidth of frequency and amplitude modulations of the note was modelled intuitively as a Gabor basis, where the window function was sinc in shape, appropriate for held-note instruments such as the bowed string and woodwind families.

The formulation of these models was deliberately chosen so that well-known and understood Bayesian priors and inference algorithms could be applied, complementing and building on existing work. The improvements in this chapter, as they are in the context of these models, may therefore be conversely applied to the original models, algorithms and applications which inspired them. For example, choosing to model the analytical representation of the observed signal revealed some improvements to the inference algorithms for these models. The damped envelope model, which can be treated as a linear dynamical system when adding Gaussian noise to the state and the observation processes, allows the posterior distribution of the partial frequencies and damping ratio to be computed in closed form if the prior is Gaussian, or otherwise a good proposal distribution to be constructed for a MCMC inference scheme. The Gabor basis model is a general linear model, with a well studied prior structure for the basis coefficients given the frequencies. We were able to derive the mode of the signal-to-noise ratio parameter in this model, which means that the amplitudes and noise variance parameters do not need to be simulated in a MCMC scheme. This reduces the computational cost and complexity of the algorithm required, and also reduces the dimensionality of the target posterior distribution.

The models and the inference algorithms developed for them were then applied to monophonic and polyphonic transcription problems. In the case of single notes playing, we focused on how many of the harmonic

partial frequencies present were detected and modelled. We saw that using the analytic representation of the signal resulted in significantly more partials being detected, and conclude that the reduction in the ambiguity of instantaneous phase and amplitude afforded by this representation is beneficial for signal model based inference methods. We also present polyphonic transcription results for the case where the number of notes is known, and compare with prior work. We found that the new models improve transcription performance for two-note mixtures when compared to prior work, but the performance for more complicated mixtures is limited by the inference algorithm's ability to accurately estimate to the extent that spurious partial detections are avoided. The benefits of increasing the accuracy of partial frequency estimation are shown in the next chapter, where we simplify the polyphonic inference algorithm to a two-stage process, estimating the partial frequencies first, and inferring the harmonic structure secondly, to improve transcription accuracy for higher number of notes, and also correctly estimate the number of notes playing in the mixture.

Chapter 6

Multiple Pitch Estimation using Non-homogeneous Poisson Processes

Point estimates of the parameters of partial frequencies of a musical note are modelled as realizations from a non-homogeneous Poisson process defined on the frequency axis. When several notes are combined, the processes for the individual notes combine to give a new Poisson process whose likelihood is easy to compute. This model avoids the data association step of linking the harmonics of each note with the corresponding partials and is ideal for efficient Bayesian inference of unknown multiple fundamental frequencies in a polyphonic mixture of notes.

6.1 Introduction

By observing the periodogram of a polyphonic mixture of notes, a trained observer can estimate the partial frequencies present in the signal from the localized peaks in the spectrum, and then suggest fundamental frequencies by observing that some of the partial frequencies are regularly spaced along the frequency axis. For example, peaks in the spectrum at 440, 880, 1320 Hz and so on suggest a fundamental frequency of 440 Hz.

In the author's experience, using the periodogram to transcribe mixtures of notes is more reliable and quicker than listening to the mixture. This method also outperforms automated transcription systems such as the signal models described in the previous chapter and state-of-the-art auditory systems, especially avoiding octave errors which plague other systems. One of the goals of this chapter is to investigate and propose models for this method in order to improve the accuracy of polyphonic transcription.

Two assumptions about the transcription process are made. The first is that the observer does not change his or her estimates of the partial frequencies when attempting to find a set of notes which fits the observations. In plain terms, the observer is trying to fit a model to the observations, incorporating errors in the partial frequency estimates into the prior, rather than fitting the observations to the model. This motivates a two-stage process where the partial frequencies are estimated first, and then a harmonic model is fitted to the frequencies. A prior model on the partial frequencies is still required however, as the observer may know the range of fundamental frequencies that can be produced by the instrument for example, but

this prior must also be defined when the number of notes in the mixture is not known.

The second assumption is that the spectral shape in the vicinity of a peak is important to the estimation of partial frequencies, whereas only the frequencies and sometimes the amplitudes of the partials are required for transcription. The spectral shape sometimes allows us to distinguish between merged harmonics of two or more notes. There are various cases where simply picking peaks of the spectrum above an adaptive noise floor is inadequate, and these cases are often the cause of transcription errors. The notes of chords in music often have overlapping harmonics, which may not be manifested as separate peaks but to the observer are obvious because of differences in spectral shape. The spectral shape also helps distinguish between noise or artifacts in the signal and genuine partial frequencies, reducing spurious detections of partials which can lead to over or under-reporting of the number of notes playing. We will use an explicit signal model with a prior on the expected spectral shape of harmonic notes to accurately estimate partial frequencies.

We do not assume that the partial estimation procedure is perfect however, and therefore need a transcription system which is capable of dealing both with missed and duplicated partial detections. The solution we present in this chapter is to use an iterative algorithm based on the signal model presented in the previous chapter to provide high quality estimates of the partial frequencies, and to model the prior on the frequency estimates as a non-homogeneous Poisson process. Choosing to use a signal model rather than a heuristic estimation scheme for the partial frequency estimation is advantageous as present and future improvements to that model will also benefit the estimation procedure here. However, it is also permissible to use other methods to estimate the partial frequencies, as was carried out previously using periodogram peak picking [Peeling et al., 2007b] and subspace methods [Peeling et al., 2007a]. In these cases, the prior on the frequencies needs to reflect the estimation procedure, for example including a uniform clutter process across the frequency axis if many spurious partials are detected.

The structure of this chapter is as follows. In Section 6.2 we introduce the properties of non-homogeneous Poisson processes and how to calculate the likelihood given a set of observed frequencies. In Section 6.3 priors for harmonic models are discussed, and suggestions for how these priors should be modified for different partial estimation methods are given. In Section 6.4 a general method for making partial estimates from a signal model is presented. Transcription results for polyphonic mixtures of notes are presented in Section 6.5 and are compared with the previous chapter and prior work. Conclusions and suggestions for future research are given in Section 6.6.

6.2 Non-homogeneous Poisson Processes

In this section we define and describe a non-homogeneous Poisson process model [Cox and Isham, 1980]. A homogeneous Poisson process is a stochastic process, defined usually over time, where the number of events $N(b) - N(a)$ occurring between time a and time b has a Poisson distribution with rate parameter λ:

$$P\left(N(b) - N(a) = k|\lambda\right) = \frac{\exp\left(-\lambda\left(a-b\right)\right)\left(\lambda\left(a-b\right)\right)^{k}}{k!}$$

λ is the expected number of events per unit of time, and is constant for a homogeneous Poisson process. A non-homogeneous Poisson process generalizes this by allow the rate parameter to vary with time.

The principal ideas behind the model are explained by considering a model based solely on the frequency

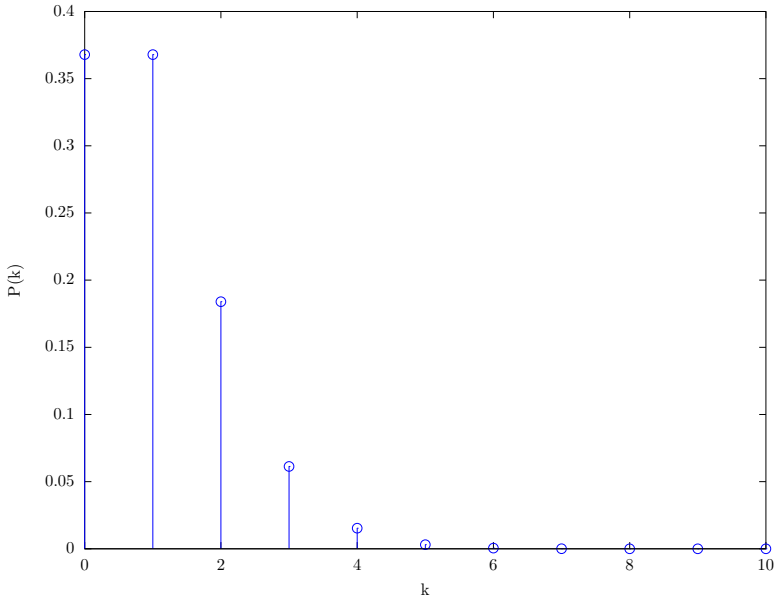


Figure 6.1: Probability mass function for the Poisson distribution

estimates of multiple partials in a musical signal.

6.2.1 Frequency-Domain Process

We define the number of partial estimates as a non-homogeneous Poisson process on the frequency axis. Let $N(f)$ be the number of partial estimates observed in the frequency range $(0, f]$. We assume that the number of partial estimates in a particular interval $(a, b]$ of the frequency axis has a Poisson distribution with parameter $\lambda_{a,b}$. The number of partials is given by $N(b) - N(a)$ and has probability distribution

$$P\left(N\left(b\right) - N\left(a\right) = k | \lambda_{a,b}\right) = \frac{\exp\left(-\lambda_{a,b}\right)\left(\lambda_{a,b}\right)^{k}}{k!} \tag{6.1}$$

We interpret $\lambda_{a,b}$ as the expected number of partials occurring in $(a, b]$. Figure 6.1 on page 82 shows the probability mass function for the Poisson distribution in (6.1) with $\lambda_{a,b} = 1$. We expect to observe one partial in the region $(a, b]$. This region could for instance be a DFT bin at a harmonic frequency of a musical note. The probability mass for observing zero and one partial in the region are equal when $\lambda_{a,b} = 1$.

Under the assumptions of a Poisson process, we write $\lambda_{a,b}$ in terms of a continuous rate function $\lambda(f)$

$$\lambda_{a,b} \equiv \int_{a}^{b} \lambda\left(f\right) \, \mathrm{d}f$$

The rate function $\lambda(f)$ of the Poisson process describes the expected concentration of partial frequencies along the frequency axis. For a harmonic musical note, we would expect the rate function to be large around the fundamental frequency and harmonics of the note, and small but non-zero elsewhere to allow for spurious

partial detections and transient effects.

For (6.1) to be valid for any values of a and b, there are two requirements. First, no two estimates may have exactly the same frequency. The signal model or partial estimation scheme used to observe partial positions should not provide estimates with exactly the same frequency, but that there must be a non-zero interval between successive frequencies. It is a property of the signal models we use in Section 6.4 that the two partials will never be estimated with exactly the same frequency, as this would lead to the basis functions being linearly dependent. Two basis functions with the same frequency may always be combined into a single basis function.

The second requirement is that the process is memoryless: the probability of a number of partials occurring in any region of the frequency axis must be independent of the occurrence of partials in any other region disjoint with that region[1]. This requires that $\lambda(f)$ contains all of the prior information about the occurrence of partials. Modelling the occurrence of partials as a Poisson process makes the model robust to missing or duplicate partial detections. Harmonic models such as described in 5.4.2 require the existence of a single partial frequency in every harmonic position modelled, and therefore an entire note may not be detected due to a single missing partial frequency.

6.2.2 Superposition

One of the key attractions of using a Poisson process model to model partial estimates is that the observation of multiple Poisson processes superimposed on the same axis is also a Poisson process. Moreover, the rate function of the combined process is formed from the summation of the individual rate functions. Formally we have M Poisson processes $N_1(f), \ldots, N_M(f)$ with rate functions $\lambda_1(f), \ldots, \lambda_M(f)$; and we observe $N_{1:M}(f) = \sum_{m=1}^{M} N_m(f)$. Then

$$\begin{aligned} P\left(N_{1:M}(b) - N_{1:M}(a) = k | \lambda_{a,b}^{(1:M)}\right) &= \frac{\exp\left(-\lambda_{a,b}^{(1:M)}\right)\left(\lambda_{a,b}^{(1:M)}\right)^k}{k!} \\ \lambda_{a,b}^{(1:M)} &= \int_a^b \sum_{m=1}^{M} \lambda_m(f)\, \mathrm{d}f \end{aligned} \tag{6.2}$$

Note that in observing $N_{1:M}(f)$ we lose labeling information, i.e., which Poisson process m each partial was generated by. This makes the likelihood (6.2) easy to compute. Inferring the actual labels of the partials, for example in a source separation setting, cannot be carried out using the superimposed process alone, however the labels may also be inferred in a probabilistic manner using a likelihood function based on the individual rate functions for each note.

6.2.3 Evaluation of Likelihood

In this section we consider how to evaluate the likelihood of the occurrence of the entire set of observed partial positions. Although we would naturally try to calculate the likelihood exactly, the method we choose depends on how we observe the Poisson process. In this section, three methods are given for evaluating the likelihood. The exact method in 6.2.3.1 should be applied when a signal model is used to estimate the

[1] This does not contradict the previous requirement that partial occurrences may not have the same frequency, as identical partial frequencies cannot be mapped to disjoint regions of the frequency axis

partial frequencies. The binning method in 6.2.3.2 is suitable when a periodogram peak picking method is employed. If the peak picking method by design only detects zero or one peaks in each frequency bin, the calculation should be modified to allow for the possibility that more than one partial frequency was present in the bin. In this case, the method in 6.2.3.3 is appropriate.

6.2.3.1 Exact Calculation

When the partial estimates are known with sufficient accuracy, and their frequencies are distinct, the likelihood of the occurrence of frequencies $f_1, f_2, \ldots, f_N$ under a non-homogeneous Poisson process $\lambda(f)$ on the frequency axis between 0 and $f_s/2$ where f_s is the sampling frequency, is given by Crowder et al. [1991], Meeker and Escobar [1998] as

$$p(f_1, f_2, \ldots, f_N, N|\lambda(f)) = \exp\left(-\int_0^{f_s/2} \lambda(f)\ \mathrm{d}f\right) \prod_{n=1}^{N} \lambda(f_n) \tag{6.3}$$

The derivation of the above likelihood is informally obtained firstly by noting that in the interval between observed frequency f_n and f_{n+1} there are no observations. Hence, using (6.2) and substituting $k = 0$, each such interval has probability $\exp\left(-\lambda_{f_n, f_{n+1}}\right) = \exp\left(-\int_{f_n}^{f_{n+1}} \lambda(f)\ \mathrm{d}f\right)$. At each observed frequency f_n, the probability, using (6.2), of observing $k = 1$ is given by $\lambda(f_n)$. We also take into account that no frequencies were observed in the interval $[0, f_1)$ and $(f_N, f_s/2]$. As a Poisson process requires that the observations in disjoint intervals of the frequency axis must be independent, we simply combine the probabilities of these observations together by multiplying them, thus:

$$\begin{aligned} p(f_1, f_2, \ldots, f_N, N|\lambda(f)) &= \exp\left(-\int_0^{f_1} \lambda(f)\ \mathrm{d}f\right) \exp\left(-\int_{f_N}^{f_s/2} \lambda(f)\ \mathrm{d}f\right) \\ &\times \prod_{n=1}^{N-1} \exp\left(-\int_{f_n}^{f_{n+1}} \lambda(f)\ \mathrm{d}f\right) \\ &\times \prod_{n=1}^{N} \lambda(f_n) \\ &= \exp\left(-\int_0^{f_s/2} \lambda(f)\ \mathrm{d}f\right) \prod_{n=1}^{N} \lambda(f_n) \end{aligned}$$

6.2.3.2 Binning

The likelihood when observations are grouped into non-overlapping regions (bins) of the frequency axis may be calculated as follows. Assume we have F such bins, spanning frequency intervals $A_1, \ldots, A_F$, and denote the number of observations in each bin by N_f. We then have, by the independence of intervals in a Poisson process,

$$\begin{aligned} P(N_1, \ldots, N_F|\lambda_1, \ldots, \lambda_f) &= \prod_{f=1}^{F} P(N_f|\lambda_F) = \frac{\exp(-\lambda_f)(\lambda_f)^{N_f}}{N_f!} \\ \lambda_f &= \int_{A_f} \lambda(f)\ \mathrm{d}f \end{aligned} \tag{6.4}$$

The advantage of this method over the exact calculation method is that the rate function λ_f may be computed in advance for each bin f before the partial frequencies are estimated, which reduces the computation required when evaluating the likelihood for multiple frames of music. Often the bins will coincide with the frequencies of the DFT used to estimate the partial frequencies.

6.2.3.3 Censored Frequencies

The partial estimation method used may only indicate that there is a partial in a frequency bin or not. An example is a single step peak picking scheme, which selects all the spectrum bins with amplitudes larger than neighbouring bins and above a noise threshold. It is possible that multiple frequencies are present within the region of the frequency axis covered by a single observation bin, for example in the case of overlapping harmonics. Although we have only 'observed' at most one frequency per observation bin, we wish to allow for the possibility that more than one frequency could be present in each bin. This is useful in practice the rate function of the Poisson process is a superposition of the rate functions of harmonically related notes. For every harmonic that overlaps within the region of a single bin, we would expect two or more partial frequencies to occur within that bin. Thus we are asserting that an observed peak in the spectrum implies the existence of multiple partial frequencies in that bin, and no observed peak implies that no partial frequencies were present in the bin.

For the observations to be valid as a Poisson process, when a peak is detected in a bin, we calculate the probability that one or more frequencies were observed in that bin, i.e., $p(N_f \geq 1) = 1 - p(N_f = 0) = 1 - \exp(-\lambda_f)$. When a peak is not detected in a bin, the probability is given by $p(N_f = 0) = \exp(-\lambda_f)$. The likelihood over all the frequency bins is thus given by

$$\prod_{f=1}^{F} \begin{cases} 1 - \exp(-\lambda_f) & \text{peak observed in bin } f \\ \exp(-\lambda_f) & \text{no peak observed in bin } f \end{cases} \tag{6.5}$$

The likelihood calculation in this case is the same as a set of Bernoulli trials with probability $1 - \exp(-\lambda_f)$.

6.3 Bayesian Priors

Bayesian inference for the non-homogeneous Poisson process involves treating the rate function $\lambda(f)$ or $\lambda(\mathbf{x}, f)$ for a vector valued process as unknown and placing a prior distribution on the rate function. A suitable choice of prior depends greatly on the observation method. If we are binning the observations into fixed, *a priori* intervals, then we can see from (6.3) that we need to infer each of the unknown parameters λ_f of the model rather than the full rate function $\lambda(f)$. However if we consider each observation and evaluate intervals between partials, then our target for inference is the rate function $\lambda(f)$.

A full non-parametric Bayesian inference of the rate function of a Poisson process is carried out in Adams et al. [2009]. A Gaussian process is used as a prior, which is transformed into a rate function using a sigmoid function. The inference is tractable, however it is not immediately clear how higher-level information, such as partials occurring at harmonic positions, could be structured into such a prior.

An interesting alternative would be to use a periodic Poisson process [Dimitrov et al., 2004]. In this model, the rate function is a periodic function along the axis. For our model, the period of the rate function

would be the fundamental frequency of the musical note.

Here however we pursue two designs of Bayesian prior which may be inferred tractably and are amenable to additional, higher level, prior structure.

6.3.1 Fixed Bins

When observations are grouped into fixed bins, then the model parameters are a finite set of positive values λ_f. Each λ_f is the intensity parameter of a Poisson distribution, for which the conjugate prior choice is the Gamma distribution:

$$p(\lambda_f) = \mathcal{G}(\alpha_f, \beta_f) \tag{6.6}$$

The posterior distribution when observing N_f is

$$p(\lambda_f | N_f) = \mathcal{G}(\alpha_f + N_f, \beta_f + 1)$$

We can integrate the unknown λ_f to obtain a negative binomial (Pascal) distribution (Figure 6.2 on page 87)

$$\begin{aligned} p(N_f) &= \frac{\Gamma(\alpha_f + N_f)}{N_f! \Gamma(\alpha_f)} p_f^{\alpha_f} (1 - p_f)^{N_f} \\ p_f &= \frac{1}{\beta_f + 1} \end{aligned} \tag{6.7}$$

Figure 6.2 on page 87 shows the prior distribution on expected number of partials λ_f (6.6) with $\alpha_f = 2, \beta_f = 1$ and corresponding marginal distribution (6.7) on observed number of partials N_f.

The hyperparameters may be optimized for the purposes of training. For example, to train the hyperparameters of the rate function for a particular musical instrument and pitch, we would use I example frames of data and estimate the partial frequencies in each frame, obtaining a set of observations $N_f^{(1)}, \ldots, N_f^{(I)}$ for each frequency bin. The posterior of the rate function given these observations is

$$p\left(\lambda_f | N_f^{(1)}, \ldots, N_f^{(I)}\right) = \mathcal{G}\left(\alpha_f + \sum_{i=1}^{I} N_f^{(i)}, \beta_f + I\right) \tag{6.8}$$

and the hyperparameters can thus be set to new values: $\alpha_f \to \alpha_f + \sum_{i=1}^{I} N_f^{(i)}$ and $\beta_f \to \beta_f + I$. The new values of the hyperparameters can now be used as the prior (6.6) for when new frames of data are observed, thus transparently incorporating training data into the Bayesian model.

6.3.2 Gaussian Mixture Model

In this section we model the entire rate function as a Gaussian mixture model (GMM). Modelling the rate function as a GMM is a convenient method to use prior information concerning the partial frequencies of harmonic instruments. The rate function is shaped by the probability density function of a Gaussian mixture

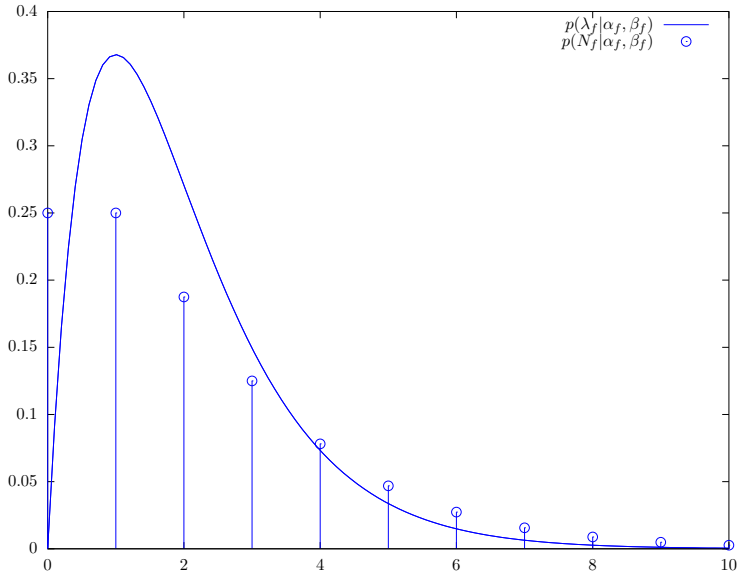


Figure 6.2: Prior on expected number of partials and marginal distribution of number of partials

model:

$$\lambda(f) = \sum_{h=1}^{H} c_h \mathcal{N}\left(\mu_h, h\sigma^2\right) \tag{6.9}$$

$$c_h \geq 0 \qquad \forall h \tag{6.10}$$

H denotes the number of mixture components. A meaningful interpretation of the above model is that H is the number of harmonic positions for a note with fundamental frequency f_0, and that a single component of the mixture corresponds to a single harmonic h. We assign

$$H = \lfloor \frac{f_s}{2f_0} \rfloor$$

where f_s is the sampling frequency. The means of the components are set to the expected harmonic positions $\mu_h \approx hf_0$ and be allowed to deviate from their ideal positions to account for inharmonicity. σ^2 allows for further spread around the harmonics, which may occur with split peaks or modulations in the signal. Finally c_h weights each harmonic, and we expect that low frequency partials have a higher probability of being detected and hence have higher values of weighting c_h.

Inference of the unknown parameters in a Gaussian mixture model involves introducing labels for each observation and using Expectation Maximization (EM). When we train our model by fitting the parameters to the estimated partial frequencies of a set of frames of audio from a harmonic musical instrument with a particular pitch, the values for σ^2 are typically small so that there is negligible overlap between the mixture components for different harmonic positions. Figure 6.3 on page 88 is provided as an example of this, where

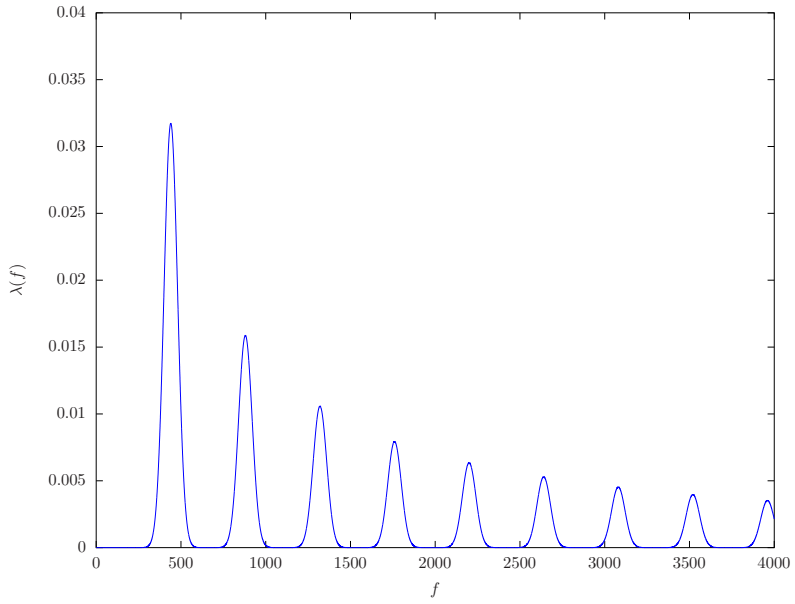


Figure 6.3: Intensity function for a musical note with fundamental frequency 440Hz parameterized using a Gaussian mixture model with $H = 9$ harmonics

we have chosen $\sigma^2 = 10^{-4}, \mu_h = hf_0$ and $c_h = 1/h$ as illustrative parameters. This assumption is also used in [Davy et al., 2006] for the model of the detuning parameters for each partial frequency. In practice, the Expectation step for training the GMM can be replaced with a K-means clustering step, where the component means are set to the expected harmonic positions, which reduces the amount of computation required for inferring the unknown parameters compared to the full Expectation step.

6.3.3 Model for mixture weights

The Gaussian mixture prior in the previous section allows for inharmonicity through the variance of each mixture component. Rather than inferring the unknown parameters of the mixture model, we can also set the parameters to particular values which match existing generative models of partial frequencies. The prior model of Godsill and Davy [2005], which is also used in Chapter 5 can be adapted as a Poisson process easily by interpreting the prior probability distribution over the number of partials and their frequencies as a counting process of the number of partials along the frequency axis. The number of partials H per note is modelled as Poisson distributed

$$p(H) = \mathcal{P}o(H|\Lambda) = \frac{\Lambda^H e^{-\Lambda}}{H!}$$

where Λ is the expected number of partials. The position of each partial frequency is normally distributed around hf_0, where h is the harmonic number and f_0 is the fundamental frequency of the note.

To convert this to a Gaussian mixture model of the form (6.9), we note that each mixture weight c_h gives the expected number of partials around hf_0. We interpret this as the probability under the generative model that the number of partials is greater or equal to h, hence following this model, the mixture weights

in (6.9) are given by

$$c_h = \left(1 - \sum_{m=1}^{h} \mathcal{P}o\left(m-1|\Lambda\right)\right) p_{\text{note}} \tag{6.11}$$

p_{note} is the prior probability that the note is playing in the mixture, and is applied as a scaling to all of the mixture weights for that model. $\sum_{m=1}^{h} \mathcal{P}o\left(m-1|\Lambda\right)$ is the cumulative Poisson distribution of observing up to $h-1$ partials. c_h, when calculated by (6.11), gives the probability of observing a partial at that frequency under the prior model.

When we set $\mu_h = hf_0$ and σ^2 to 3×10^{-8} we obtain the multiplicative inharmonicity model suggested in Godsill and Davy [2005].

6.4 Signal Model Based Partial Estimation

In previous work using non-homogeneous Poisson processes for polyphonic transcription, we have used several methods for extracting the partial frequencies as a preprocessing step. In Peeling et al. [2007b] a heuristic scheme for selecting peaks in the Fourier spectrum above an adaptive noise threshold was considered. Such a scheme is quick and can detect the partial frequencies of high amplitude notes without difficulty. However without an explicit signal model this scheme cannot differentiate between genuine partials and transient features of the noise floor, and its performance is therefore limited.

In Peeling et al. [2007a] a matrix pencil scheme for estimating damped sinusoids was used to provide frequency and amplitude data for partials. Matrix pencil schemes use eigenvalue analysis to decompose the signal into a number of sinusoids, from which partial-frequency estimates can be obtained. This is an improvement as an explicit signal model is being used, but the number of sinusoids needs to be supplied to the estimation scheme. As this is not known *a priori*, the number of sinusoids has to be estimated separately. Underestimating or overestimating the number of sinusoids can result in frequency estimates that differ greatly from the true frequencies, leading to transcription errors, as the algorithm attempts to fit an incorrect number of sinusoids to explain the data and adapts the frequencies accordingly.

The most satisfying partial estimation scheme we have investigated and present here is to apply an existing probabilistic signal model, and to iteratively estimate the number of partials and their frequencies accurately using a Bayesian model selection criterion for that signal model to determine when a suitable number of partials have been detected. The scheme is similar in structure to Matching Pursuit [Mallat and Zhang, 1993] which at each iteration selects a basis function from an over-complete dictionary of basis functions such as Gabor functions (5.2.5) to match the residual of the signal. The procedure here differs from Matching Pursuit in the following aspects: a set of basis functions with the same frequency are selected per iteration; a local search is employed to identify a suitable frequency at each iteration, using the periodgram as an initial estimator (6.3.3) and additionally numerical optimization to minimize the residual (6.4.3), rather than a global search over the dictionary; and the g-prior is incorporated in the signal model which improves the correct selection of the number of partials when using Bayesian model selection.

The frequency estimates obtained are much less sensitive to the modelled number of partials than for the matrix pencil scheme. Moreover, future improvements in these signal models will also improve the quality of the frequency estimates. Using a signal model selection criterion requires much more computation than other schemes, but substantial improvements in transcription accuracy are obtained as a result.

Although we describe the algorithm making reference to the probabilistic signal model developed in Chapter 5, in practice any suitable signal model with an accompanying model selection criterion can be used for the partial estimation scheme component of this system.

6.4.1 Overview

To estimate partial frequencies using a signal model, we use a simple iterative approach. At the beginning of an iteration, we have a set of partial frequencies, and using the signal model, we can assign some of the signal to the model, and the remainder to a residual. The residual contains the remaining partials that we are yet to detect. A model selection criterion is used to calculate how well the current set of partial frequencies explains the observed signal. We then select a new frequency from the residual, and add this to the set of partial frequencies. When the model selection criterion fails to improve by adding partial frequencies, the algorithm is stopped.

In general, the best frequency to select from the residual is that which accounts for the most energy in the residual. This approach naturally will tend to reduce the number of partials that are ultimately selected. Selecting the maximum of the Fourier spectrum is clearly a good estimate of this frequency. We also investigated using the maximum as a starting point, and using numerical methods to maximize the energy removed from the residual. For a probabilistic signal model, this is equivalent to finding the MAP frequency estimate [Bretthorst, 1989].

6.4.2 Bayesian Model Selection Criterion

To exactly estimate the number of partials using a probabilistic signal model requires a reversible jump MCMC scheme, as covered in the previous chapter. However we have found that using a Bayesian model selection criterion, which is an approximate means to compare models of different dimensions, produces acceptable estimates of the number of partials. In Djuric [1996, 1993] a model selection criterion for the estimation of complex valued sinusoids in white noise was derived:

$$N \log \left(y^{\top} \mathbf{P} y \right) + \frac{5k}{2} \log N + \log p\left(\theta\right) \tag{6.12}$$

where k is the number of sinusoids and $\mathbf{P}$ has the same definition as in (5.23). $p\left(\theta\right)$ is the prior probability of any additional model parameters. The appropriate number of sinusoids is that which minimizes the above expression.

In the remainder of this chapter, we will use the Gabor model, using sinc basis functions, as described in 5.3.4. The only additional model parameter that it is necessary to infer is the expected signal-to-noise ratio ξ which is assigned an inverse-Gamma prior $p\left(\xi\right) \sim IG\left(\xi; \alpha_{\xi}, \beta_{\xi}\right)$ where $\alpha_{\xi} = 2$ and $\beta_{\xi} = 1$ are chosen to give this prior infinite variance and thus be uninformative.

We divide the signal into frames with 50% overlapping samples. The partial frequencies are estimated in each frame separately. At each iteration we select and add a frequency to the set of partial frequencies already estimated for that frame. If this addition increases the model selection criterion (6.12), then we terminate at this point. Otherwise we update all of the estimated frequencies, and continue. The scheme is described in more detail in Algorithm 6.1.

Algorithm 6.1 Partial estimation scheme for a frame of audio y with N samples

- Initialize: $k \leftarrow 0$, $M_0 = \frac{N}{2} \log\left(y^\top y\right)$,$r = y$
- Iterate while $M_k \geq M_{k-1}$,
  - $k \leftarrow k + 1$
  - Estimate new partial frequency ω_k from the residual r by the method in 6.4.3 or 6.4.4
  - Form the basis matrix: $\mathbf{D}_{t,k} = \exp i\omega_k t$, $t = 1, \ldots, N$
  - Estimate signal to noise ratio, $\xi^* = \arg\max_\xi p\left(\xi|y\right)$ and $\mathbf{P}$
  - $M_k = \frac{N}{2} \log\left(y^\top \mathbf{P} y\right) + \frac{5}{2} k \log N + \log p\left(\xi^*\right)$
  - Calculate partial amplitudes: $\mathbf{b} = \left(1 + \frac{1}{\xi^*}\right)^{-1} \left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \mathbf{D} y$
  - Update residual of signal $r \leftarrow y - \mathbf{D}\mathbf{b}$

An important step in Algorithm 6.1 is the method by which the frequency value of a new partial is estimated from the residual of the signal. In the remainder of this section, we investigate two methods for producing an accurate frequency estimate from the residual of the signal at each iteration of the algorithm. In Section 6.1 we stated that the prior on the partial frequency estimates practically depends on the estimation scheme. For each method therefore, we present a rate function for single harmonic notes. For polyphonic mixtures of harmonic notes, the rate functions can be superimposed (see 6.2.2). In Section 6.5 we describe inference of polyphony using these rate functions, and compare the accuracy of both methods within the partial estimation scheme for polyphonic music transcription.

6.4.3 Zero-Padding

The first method we investigated to estimate the value of a partial frequency in the residual was to zero pad the residual to a length of $4N$ and find the maxima of the DFT spectrum. Zero padding interpolates the DFT spectrum, increasing the number of discrete frequencies at which the partial frequency can be found. An example of output of the partial estimation scheme in Algorithm 6.1 for a polyphonic note mixture is provided in Figure 6.5 on page 94. We see that many of the partials visible in the spectrum of the signal are detected over multiple frames with only a small number of additional spurious detections. Based on our observation of the partial estimation results, we propose a novel parametric rate function which is designed to robustly infer multiple fundamental frequencies when estimating partial frequencies iteratively from the DFT spectrum. As the DFT spectrum gives discrete estimates of the frequency in bins, the likelihood function of the observed partial frequency estimates should be calculated using the method described in 6.2.3.2.

The rate function we propose has the following form:

$$\lambda_f\left(f_0\right) = \begin{cases} \lambda_{\text{Note}} & |f/f_0 - [f/f_0]| < \epsilon \\ \lambda_{\text{Clutter}} & \text{otherwise} \end{cases} \tag{6.13}$$

where ϵ be the maximum allowed inharmonicity and f is the central frequency of the DFT bin. The notation $[f/f_0]$ denotes rounding to the nearest integer, and hence gives us the position of the closest harmonic of f_0 to

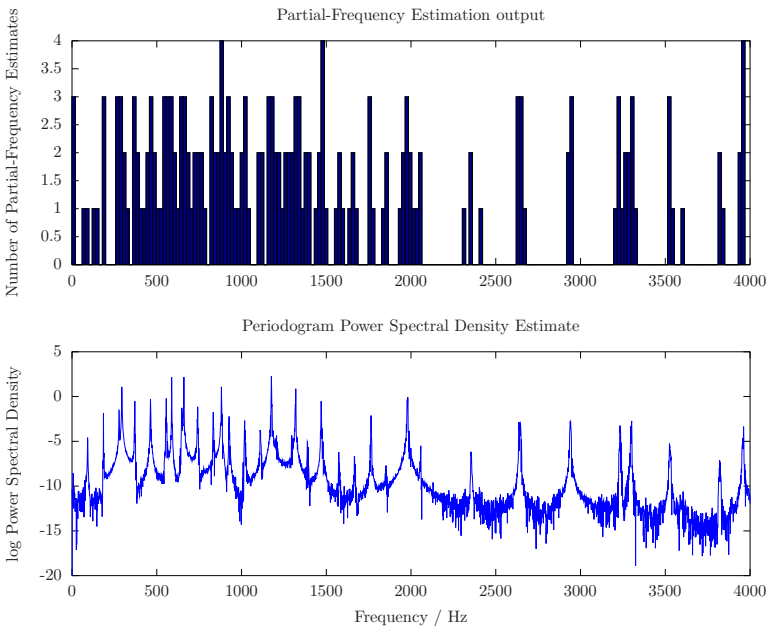


Figure 6.4: Partial estimation results for zero padding method

the central frequency f. ϵ allows for multiple partials per harmonic, which can occur because of modulations in the signal, or because of modelling and estimation error. This inharmonicity model is multiplicative, allowing for a larger deviation from the ideal position for the higher frequency harmonics. The rate function bears some resemblance to the scoring function (3.2) for inharmonicity used by Yeh et al. [2005]. This intensity function approximately resembles histogram data of partial estimates for the low harmonics, see for example Figure 6.4 on page 92. The partial estimation scheme described in Algorithm 6.1 returns clusters of partials around the harmonic positions, and a few spurious estimates. We capture this behaviour with a parametrized intensity function (6.13).

Our expectation is that one partial in the correct position would be detected in each frame of data, so we set λ_{Notes} equal to the number of frames. λ_{Clutter} models additional detections and must be greater than zero in each frequency bin. λ_{Clutter} should be independent of the actual notes being played, and may be estimated or inferred. However we found that the model is quite robust to a range of clutter values, and we set $\lambda_{\text{Clutter}} = 0.1$ in the following experiments.

The inharmonicity parameter ϵ is also unknown, and depends on the collection of notes being played. Instruments with high modulations (such as vibrato) will have large values of ϵ as well as inharmonic instruments. We assign a uniform prior $p(\epsilon) = U\left(\frac{1}{100}, \frac{1}{10}\right)$ and infer ϵ with each observation. Note that λ_f has a functional dependence on ϵ hence it is unknown, and we use the Bayesian prior described in 6.3.1 to correctly calculated the appropriate marginal likelihood value.

6.4.4 Likelihood-Search

The second method we investigate is to directly maximize the likelihood of the Gabor signal model when adding a new frequency estimate to the set of partial frequencies estimated in the residual. This is motivated by our observations in 5.5.2 where we found that there could be significant deviation between the maxima of the DFT spectrum and the maximum likelihood frequency estimate, and that the residual of the signal after subtracting the detected frequency could contain additional peaks which can result in spurious partial frequency detections, as shown in Figure 5.3 on page 77.

Maximization of the likelihood which is equivalent to minimizing the energy of the residual signal was carried out numerically, using the Golden search method. The steps for calculating the residual are shown in Algorithm 6.1. This method requires an upper and lower bound for the frequency, between which the maxima is estimated. First we chose the maxima of the Fourier spectrum as an initial estimate, as in 6.4.3 but without any zero padding. The bounds for the Golden search method were chosen to be $\pm$10Hz of the initial estimate, based on our observations of the discrepancy between the Fourier spectrum and the signal model maxima in 5.5.2.

An example of the partial estimation results are shown in Figure 6.5 on page 94. These results show that the partial estimates are very good, in that many of the partials are detected, and there are very few duplicate or spurious estimates. As these results match the actual partials in a harmonic note, we simply use the rate function described in 6.3.3 which was derived from a Bayesian harmonic model.

This method requires more computation per iteration than the method in 6.4.3 but as less spurious and duplicate partials are detected, the overall number of iterations is smaller, and the cost of computing the model selection criteria is also reduced. We found that the likelihood search method was quicker overall than the zero padding method. As it also produces better partial estimation results which can then be analyzed using an acceptable Bayesian model and compared easily with other inference schemes, we recommend the likelihood-search over the zero-padding method from a modelling and practical point of view. As we shall see in Section 6.5, there are also mild improvements in polyphonic transcription performance.

6.5 Polyphonic Pitch Estimation

The Poisson process model is useful for polyphonic pitch estimation because the likelihood is quick to compute, and it is therefore feasible to perform searches over pitch candidates, exhaustively testing every single note and also pairs of note pitches, at each state selecting the single note or pair of notes that results in the highest likelihood.

The performance of the model is very much dependent on the quality of the partial frequency estimates, although the Poisson process model allows for more clutters, errors and inaccuracy than direct inference using a poor signal model would.

6.5.1 Greedy Search

In this section we present results for a maximum marginal likelihood approach comparing the two partial estimation schemes described in Section 6.4. Following other inexpensive multiple pitch estimation schemes, we search in a greedy manner, adding one note at a time to the mixture, selecting the maximum likelihood

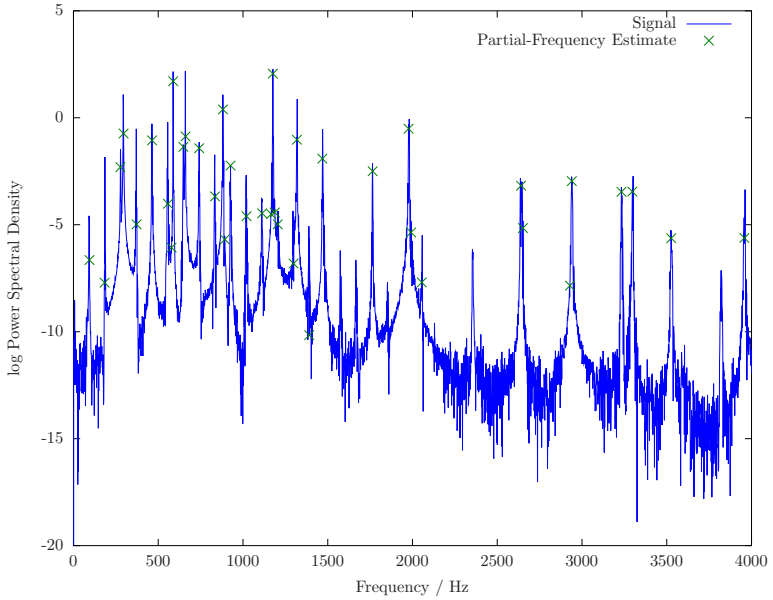


Figure 6.5: Partial estimation results and periodogram estimate for a polyphonic mixture of four notes.

solution at each point. For the zero-padding method, the observations are observed in bins, hence the likelihood function is given by (6.4), whereas for the likelihood-search method, the observed frequencies are known precisely, hence the likelihood function is given by (6.3).

We evaluate the model using the dataset in Davy et al. [2006], described also in 5.5.2 and compare with the results obtained there. The polyphonic note mixtures are buffered into frames of length 1024 samples with 50% overlap. Table 6.1 on page 95 present our results for when the number of notes is known in advance, and are compared to the results in Table 5.3 on page 76 on the same data set, showing that we are able to achieve a comparable and even superior level of performance to a full Bayesian model and also a state-of-the-art auditory model. The likelihood search method in general makes fewer transcription errors than the zero-padding method.

In these results, we observe that the Poisson process model is capable of correctly inferring multiple notes with the same pitch. Due to the superposition property of Poisson processes, it is straightforward to test whether a greedy search performs worse than a more exhaustive search which also considers adding pairs of notes to the solution at each iteration. We can also check whether the search leads to a higher likelihood solution than the likelihood of the true pitches under this model. For the results in Table 5.3 on page 76 the greedy search returns the same sets of notes as the method which adds pairs of notes, and the performance of the search methods are identical. Thus the greedy algorithm is shown to be sufficient in finding the maximum likelihood solution, and we suggest that this is a property that results from the superposition property.

		Notes in Mixture			
Evaluation Metric	Model	1	2	3	4
% octave error	Zero-padding	0	4.9	11.9	8.9
	Likelihood-search	0	7.0	6.0	7.5
	Gabor	0	2.8	11.1	10.2
	Davy et al. [2006]	0	10.3	17.8	9.3
	Klapuri [2008]	0	13.6	19.0	22.2
% pitch error	Zero-padding	0	6.7	9.0	17.0
	Likelihood-search	0	5.0	10.7	15.5
	Gabor	0	8.3	15.6	18.6
	Davy et al. [2006]	0	5.1	7.2	19.7
	Klapuri [2008]	0	1.4	6.0	10.3
% total error	Zero-padding	0	11.6	20.9	25.9
	Likelihood-search	0	12.0	16.7	23.0
	Gabor	0	11.1	26.7	28.8
	Davy et al. [2006]	0	15.4	25.0	29.0
	Klapuri [2008]	0	15.0	25.0	32.5

Table 6.1: Polyphonic pitch estimation using the Poisson process model, comparing the zero-padding and likelihood-search methods for estimating the frequency of the partial at each iteration of the partial estimation scheme. The results are compared to estimation results for the Gabor model developed in Chapter 5, the Bayesian harmonic model of Davy et al. [2006], and a state-of-the-art auditory model for polyphonic transcription [Klapuri, 2008].

6.5.2 Estimation of number of notes

The greedy search method may also be used to estimate the number of notes. Once the partial frequencies have been estimated, as a result of the superposition property there are no remaining parameters in the model. There is no danger of overfitting too many notes using the Poisson process model once the partial frequencies have been estimated, as the expected number of partials in the signal is implicitly defined by the integral of the rate function of the Poisson process. Attempting to fit too many notes will result in a much higher expected number of partials than actually observed, which is penalized by the likelihood of the Poisson process. This is advantageous, as there is no need for an explicit penalization term to avoid overfitting of the model. The likelihood itself is therefore sufficient for determining the number of notes. Notes are added to the candidate set until this fails to increase the likelihood.

We evaluate the estimation of the number of notes by calculating the average precision and recall for different numbers of notes and overall, and present our results in Table 6.2 on page 96. The precision P is defined as the number of correct notes detected in the mixture divided by the estimated number of notes by the system. The recall R is defined as the number of correct notes detected in the mixture divided by the actual number of notes in the mixture. Both precision and recall are commonly expressed as percentages. The F-Score F, given by

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

is in excess of 80% for both methods, showing that the number of notes is being estimated accurately.

Overall the two partial estimation schemes, with their associated rate functions, have the same level of accuracy for polyphonic transcription. The zero-padding method overall has a higher recall than the

		Notes in Mixture				
Evaluation Metric	Model	1	2	3	4	Overall
Precision %	Zero-padding	93.8	85.4	72.9	68.4	75.7
	Likelihood-search	98.0	85.0	76.7	71.0	78.2
	Klapuri [2008]	87.0	73.9	65.2	58.7	66.2
Recall %	Zero-padding	100.0	91.8	86.7	87.7	88.5
	Likelihood-search	100.0	92.4	85.8	87.1	85.9
	Klapuri [2008]	100.0	85.0	75.0	67.5	76.5
F-Score %	Zero-padding	96.8	88.5	79.2	76.9	81.6
	Likelihood-search	99.0	88.5	80.1	78.2	81.9
	Klapuri [2008]	93.0	79.0	69.8	62.8	71.0

Table 6.2: Precision and recall for estimating the number of notes and their pitches using the Poisson process model and the two frequency estimation schemes described in this chapter. The results are compared to a state-of-the-art auditory model.

likelihood-search method, which is expected as the zero-padding method returns more partial frequencies, hence potentially detecting more notes.

In our opinion and experience, a higher recall is preferable for an offline polyphonic transcription system, where the results can be verified by a trained musician, as it is perceptually simpler to notice and delete a spurious, incorrectly transcribed note than it is to determine and add a missing note. However, we would still prefer to use the likelihood-search method as it requires less computation in the partial estimation stage. To increase the recall of the likelihood-search method, the $5k/2$ penalization term in (6.12) can be reduced, so that more partials are detected in the estimation scheme, and thus more notes are detected. Even when reducing the penalization term, the number of partials returned by the likelihood search method is substantially less than returned by the zero-padding method, and the computational savings are retained.

6.5.3 Comparison with State-of-the-Art

In this section we compare the performance of the system developed in this chapter with other state-of-the-art multiple pitch estimation systems that compute estimates on a frame-by-frame basis. The results we compare with are taken from Vincent et al. [2010] using the MIREX 2007 woodwind training dataset[2]. Test excerpts are generated by successively summing together the first 30 seconds of the flute, clarinet, bassoon, horn and oboe tracks in order. To use the same evaluation criterion, we use overlapping frames of 46ms length spaced 10ms apart. The results are presented in Table 6.3 on page 97 for the best performing algorithms evaluated in Vincent et al. [2010], and also the likelihood-search and zero-padding methods described here.

The results demonstrate that the two multiple pitch estimation schemes of Vincent et al. [2010] outperform the likelihood-search method on this woodwind dataset, although the performance of the likelihood-search method is comparable with state-of-the-art systems. There are also significant differences between the results on this real dataset and the results on the isolated examples presented in Table 6.2 on page 96, where the likelihood-search and zero-padding methods have a similar level of performance, show significant improvement over the system of Klapuri [2008] and have a higher F-measure overall than in Table 6.3 on page 97. These differences are clearly due to the number of frames per note estimate used. The isolated

[2] `www.music-ir.org/mirex2007/`

	Polyphony			
Algorithm	2	3	4	5
Unconstrained NMF Vincent et al. [2010]	79.9	56.3	62.1	61.9
Constrained NMF Vincent et al. [2010]	76.5	64.7	67.5	62.5
Klapuri [2008]	73.4	59.1	63.5	59.9
Likelihood-search	75.0	66.1	55.0	59.6
Zero-padding	66.2	59.4	51.2	56.3

Table 6.3: Comparison of the F-measure of multiple pitch estimation for the MIREX 2007 woodwind training dataset. The 'Constrained NMF' algorithm is known as 'NMF under harmonicity and spectral smoothness constraints' in Vincent et al. [2010].

examples, which have several frames per note, are realistic only when concurrent sounding note segments are extracted from the audio beforehand, whereas each frame in the woodwind dataset contains much less spectral information. The zero-padding method suffers especially for short frames as the frequency estimates are only obtained from the spectral information alone, whereas the likelihood-search method is able to obtain more accurate estimates by fitting a signal model to the audio. Additional improvements could be made by jointly inferring the parameters of the prior harmonic model described in 6.3.3, as the parameters were chosen in that model to be suitable for longer notes. Alternatively, larger improvements could be made by extending the signal model with dependencies between neighbouring frames, rather than independently inferring isolated frequencies and notes in each frame. This would thus effectively increase the length of the frame available for estimating the frequencies, thus improving the accuracy of the estimates in line with the results presented in Table 6.2 on page 96.

6.6 Conclusion

In this chapter we have motivated and implemented a two-stage process for polyphonic pitch detection. The first stage is to accurately detect partial frequencies in a short segment of the signal. The second stage is to infer the notes using a harmonic model of the expected frequencies. The approach we have used is to adapt existing Bayesian signal and prior harmonic models for musical notes and design simple, approximate inference schemes for these models which are computationally inexpensive and allow for the exploration of many more combinations of notes than may be feasible using full Bayesian models. As a result we have shown that a higher level of transcription accuracy can be achieved even with a simple algorithm design. Moreover we are able to benefit from present and future advances of the Bayesian models to improve partial frequency and multiple estimation.

The partial frequency estimation stage is carried out by progressively fitting a sinusoidal basis to the observed signal, halting the process when a Bayesian model selection criterion fails to improve by adding more partial frequencies. There are a number of requirements on the signal model for this method to be feasible, firstly that the optimum frequency to be added per iteration can be found quickly, and secondly that adding additional frequencies to the model does not require that the existing frequencies be modified. These requirements are met by the Gabor basis models discussed in Chapter 5 but are not met by matrix pencil methods which were used in previous work [Peeling et al., 2007a].The primary benefit of using explicit signal and noise models over heuristic peak-detection schemes is that the spectral shape of partials and the

level of the noise floor can be used to distinguish between actual partials and spurious artefacts, and also detect partials with small amplitudes which are otherwise masked by nearby partials with larger amplitudes.

The estimated partial frequencies are then tested against a series of multiple pitch hypotheses by evaluating the likelihood of the frequency positions under the assumption of a non-homogeneous Poisson process. We choose to use a Poisson process for the following reasons: it is a generative model for which Bayesian priors for different harmonic models and partial frequency estimation schemes can be created easily; the likelihood can be calculated exactly without the use of an iterative algorithm, and the superposition property means that multiple notes can be inferred using a greedy search scheme where previously found notes are consistent with the current hypothesis, and the number of notes can be estimated correctly.

The resulting transcription scheme is accurate for isolated notes, although the accuracy when used to transcribe music frame-by-frame for short frame lengths is much less than for isolated notes. Future work should concentrate on extending the signal model to include dependencies between neighbouring frames, to allow the frequencies to be estimated in each frame with much higher accuracy than possible for short frame lengths.

The relationship between the pitches in neighbouring frames, although not explored in this chapter, can be expressed through prior, generative models of pitch trajectories and note durations. For example, the transcription system here could be used to evaluate the likelihood of observed partial frequencies in each frame for different note combinations, and the Viterbi algorithm used to infer the most likely transcription across multiple frames using a hidden Markov model prior. Multiple passes would be used to add additional notes to each frame, similar to the greedy search procedure presented here. These ideas are developed further in Chapter 8.

Although the method we have presented here is accurate and flexible, it is not suitable for large scale processing of musical data as the algorithms for both frequency and pitch estimation are iterative in nature, and their computation cannot be easily parallelised to make use of optimized software libraries or parallel processing in hardware. In the next chapter we develop and apply alternative Bayesian generative models where the signal is projected onto a fixed set of harmonic bases, one per pitch, rather than attempting to infer the basis of the signal model. The relative amounts of energy used in each projection are used to create a transcription of long passages of polyphonic music. This approach allows multiple frames and the dependencies between them to be processed in parallel.

Chapter 7

Gaussian Variance Generative Matrix Factorization Models

In this chapter we develop a generative Bayesian model for matrix valued observations, where each element of the matrix is assumed distributed zero mean Gaussian, and the variances of the elements are factorized into two positive-valued matrices with smaller common dimension than those of the observed matrix. The models represent the observed data as the superposition of statistically independent sources. These models can be used for dimensionality reduction, modelling and compression of real-valued data, such as the short-time Fourier transform (STFT) of an audio signal, where it can be applied to source separation and music transcription by jointly modelling spectral characteristics such as harmonicity and temporal activations or excitations. Suitable priors are chosen so that the appropriate modelling dimensionality is selected, and the variance parameters can be inferred using efficient matrix update equations, allowing large amounts of data to be processed efficiently. The algorithm is adapted for the task of polyphonic transcription of music using labeled training data. The performance of the system is compared to that of existing discriminative and model-based approaches on a dataset of classical piano music.

7.1 Introduction

Tools for multivariate data analysis, processing and compression include principal components analysis (PCA) and non-negative matrix factorization (NMF), which perform dimensionality reduction. Recently these tools have been made more flexible and powerful by description in a probabilistic, statistical framework, using a generative model. Existing algorithms can then typically be described as performing maximum likelihood estimation of the parameters. Tipping and Bishop [1999] describe PCA in a probabilistic framework as a Gaussian latent variable model, and use the Expectation-Maximization (EM) algorithm to iteratively reach a solution. Virtanen et al. [2008] describe NMF using a Poisson source model, and obtain the iterative update equations for the information-divergence measure given by Lee and Seung [2000]. The advantages cited by adopting a probabilistic signal model are the ability to incorporate prior information via Bayesian methods, and a consistent approach to dealing with multiple observations and missing data. See also Cemgil [2008] for a full report on a Bayesian NMF model with applications to missing data.

Non-negative matrix factorization has been used with some success in the modelling of time-frequency energy distributions in audio and musical signal applications, such as drum transcription [Paulus and Virtanen, 2006], source separation [Wang and Plumbley, 2005, Virtanen, 2007], and polyphonic music transcription [Smaragdis and Brown, 2003, Bertin et al., 2009a, Abdallah and Plumbley, 2004].

To illustrate the principal concept, consider a matrix

$$\mathbf{X} = \left|\{x^2_{\nu,\tau}\}\right|$$

formed of the coefficients of the short-time Fourier transform of an audio signal (see Section 7.4) with frequency indices $\nu = 1, \ldots, F$ and time indices $\tau = 1, \ldots, T$. This non-negative matrix can be approximately decomposed into two non-negative matrices

$$\mathbf{X} \approx \mathbf{TV}$$

$\mathbf{T}$ is a $F \times I$ matrix which is typically interpreted as a set of harmonic *template* vectors $[\mathbf{t}_1, \ldots, \mathbf{t}_I]$ along the frequency axis, and $\mathbf{V}$ is a $I \times T$ matrix, interpreted as a set of activation or *excitation* vectors $[\mathbf{v}_1, \ldots, \mathbf{v}_I]^\top$ in time. In a single channel source separation situation, the observed matrix $\mathbf{X}$ can be written as the superposition of I sources:

$$\begin{aligned} \mathbf{X} &\approx \sum_{i=1}^{I} \mathbf{t}_i \mathbf{v}_i^\top \approx \sum_{i=1}^{I} \mathbf{S}_i \\ \mathbf{S}_i &= \mathbf{t}_i \mathbf{v}_i^\top \end{aligned}$$

The representation of $\mathbf{X}$ as the sum of a set of single rank matrices is shown schematically in Figure 7.1 on page 101. However, this model is physically unrealistic: because energy is a quadratic quantity, the energy for two sources is not additive, i.e.,

$$|s_1 + s_2|^2 \neq s_1^2 + s_2^2$$

This is typical of spectral modelling where superposition is not addressed. Here we describe a probabilistic model based on the transform coefficients themselves rather than a positive energy representation. The source model is conditionally zero-mean Gaussian, motivated by the underlying physics, which obeys the superposition property desired. The model is valid for real-valued observations which arise from the discrete Cosine transform (DCT) for instance, and also complex-valued coefficients from the discrete Fourier transform (DFT). The statistical perspective readily admits a Bayesian framework in the form of prior distributions placed on the variance parameter of the Gaussian. A review of existing prior structures and related inference methods is in Godsill et al. [2007].

This chapter is an expansion of Peeling et al. [2010] which develops an Expectation-Maximization (EM) algorithm for the Gaussian variance model. Here we additionally discuss variational Bayes and Monte-Carlo inference techniques in Section 7.3, and demonstrate additional applications to musical audio processing in Section 7.4 other than polyphonic music transcription.

In this chapter, we use the Gaussian variance model in a matrix factorization framework. We first express the model and obtain the maximum likelihood estimation of the factorization as an iterative EM algorithm, and describe the implementation as a pair of matrix-update equations to demonstrate that the Gaussian model shares the same computational attractiveness as related NMF approaches (Section 7.2). We then

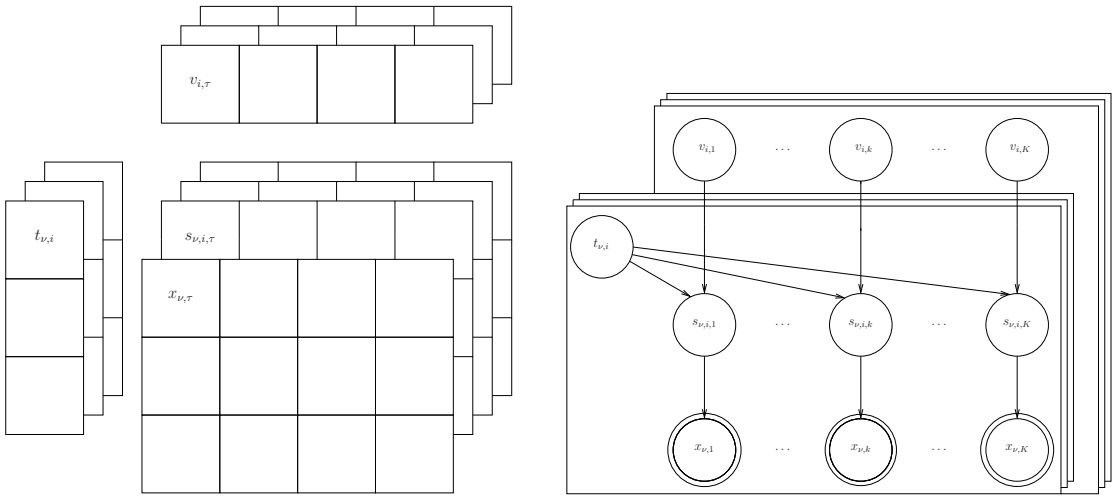


Figure 7.1: Representations of the single-channel source separation model as a matrix factorization problem

place conjugate priors on the elements of the factor matrices, and describe a number of Bayesian inference techniques: variational Bayes and a set of Monte-Carlo techniques (Section 7.3). We present demonstrations of applications for this model in the field of musical audio processing in Section 7.4, and by placing a prior model for MIDI transcription (Section 7.5) we develop a system for polyphonic transcription of piano music. We present comparative results on a large dataset of music in Section 7.6.

7.2 Gaussian Variance Matrix Factorization Model

In this section we describe a matrix factorization model, where the observed matrix coefficients have a zero mean Gaussian distribution, where the variance of each coefficient is obtained from the matrix product of the template and excitation matrices. This model was used in single channel audio source separation by Benaroya et al. [2003] and polyphonic music transcription by Abdallah and Plumbley [2004], and was linked with the Itakura-Saito (IS) divergence between the observed matrix coefficients and the underlying variances by Févotte et al. [2009].

We initially express the model as a probability distribution over individual sources.

$$\begin{aligned} s_{\nu,i,\tau} &\sim \mathcal{N}(s_{\nu,i,\tau}; 0, t_{\nu,i} v_{i,\tau}) \\ x_{\nu,\tau} &= \sum_i s_{\nu,i,\tau} \end{aligned} \tag{7.1}$$

The s variables represent the individual *latent sources*, and the x variables are the observations, formed from the superposition of the sources. When the latent source variables and the observed coefficients are complex valued, the distribution in (7.1) is the circular symmetric complex normal distribution (Section A.1) i.e., the real and imaginary parts are uncorrelated and have equal variance.

The matrix representation of the superposition is

$$\mathbf{X} = \mathbf{S}_1 + \cdots + \mathbf{S}_I = \sum_{i=1}^{I} \mathbf{S}_i$$

where $\mathbf{X}, \mathbf{S}_i, i = 1, \ldots, I \in \mathbb{R}^{F \times T}$ have elements $x_{\nu,\tau}, s_{\nu,i,\tau}$, for $\nu = 1, \ldots, F$, $\tau = 1, \ldots, T$ respectively. Marginalizing out the latent sources $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_I\}$ gives

$$p(\mathbf{X}|\mathbf{T}, \mathbf{V}) = \int \mathrm{d}\mathbf{S}\, p(\mathbf{X}|\mathbf{S}) p(\mathbf{S}|\mathbf{T}, \mathbf{V}) = \prod_{\nu,\tau} \mathcal{N}\left(x_{\nu,\tau}; 0, \sum_i t_{\nu,i} v_{i,\tau}\right)$$

due to the superposition property of normal random variables, that is: when

$$\begin{aligned} s_i &\sim \mathcal{N}\left(s_i; 0, \sigma_i^2\right) \\ x &= s_1 + \cdots + s_I \end{aligned}$$

then the marginal probability is given by

$$p(x) = \mathcal{N}(x; 0, \sum_i \sigma_i^2)$$

For real x, the marginal log-likelihood of a single observation is given by:

$$\log p(\mathbf{X}|\mathbf{T}, \mathbf{V}) = \sum_{\nu} \sum_{\tau} \left(-\frac{1}{2\sigma_{\nu,\tau}^2} x_{\nu,\tau}^2 - \frac{1}{2} \log 2\pi\sigma_{\nu,\tau}^2 \right) \tag{7.2}$$

and for complex x

$$\log p(\mathbf{X}|\mathbf{T}, \mathbf{V}) = \sum_{\nu} \sum_{\tau} \left(-\frac{1}{\sigma_{\nu,\tau}^2} |x_{\nu,\tau}|^2 - \log \pi\sigma_{\nu,\tau}^2 \right) \tag{7.3}$$

where

$$\sigma_{\nu,\tau}^2 = \sum_i t_{\nu,i} v_{i,\tau} = [\mathbf{TV}]_{\nu,\tau}$$

The derivation for the real and complex valued models is so similar that when it is required to specify which observation model is being used, we let $D = 1$ for the real valued model, and $D = 2$ for the complex valued model. This is motivated by viewing the complex normal distribution as a two-dimensional normal distribution with equal variance on the real and imaginary axes. The marginal log-likelihood in its general form is thus

$$\log p(\mathbf{X}|\mathbf{T}, \mathbf{V}) = \sum_{\nu} \sum_{\tau} \left(-\frac{D}{2\sigma_{\nu,\tau}^2} |x_{\nu,\tau}|^2 - \frac{1}{D} \log D\pi\sigma_{\nu,\tau}^2 \right) \tag{7.4}$$

As observed in Févotte et al. [2009], maximizing the log-likelihood is equivalent to minimizing the IS divergence

$$d_{\mathrm{IS}}\left(z|\sigma^2\right) = \frac{z}{\sigma^2} - \log z + \log \sigma^2 - 1$$

between $z = \left|x^2\right|$ and σ^2. This can be seen by comparing (7.3) and (7.4) and ignoring the elements of both equations that do not depend on the variances σ^2.

7.2.1 Maximum-likelihood and the EM algorithm

The EM algorithm for maximum-likelihood estimation of parameters in the Gaussian variance model was independently derived by Févotte et al. [2009]. Maximizing the likelihood of the Gaussian variance model is equivalent to minimizing the Itakura-Saito distance [Itakura and Saito, 1968] between the observed matrix $\mathbf{X}$ and its reconstruction $\mathbf{TV}$.

The log likelihood of observed datum $\mathbf{X}$ can be written as

$$\begin{aligned}\mathcal{L}_{\mathbf{X}} &\equiv \log \int \mathrm{d}\mathbf{S}\, p(\mathbf{X}, \mathbf{S}|\mathbf{T}, \mathbf{V}) \\ &= \log \int \mathrm{d}\mathbf{S}\, \frac{q(\mathbf{S})}{q(\mathbf{S})} p(\mathbf{X}, \mathbf{S}|\mathbf{T}, \mathbf{V}) \\ &\geq \int \mathrm{d}\mathbf{S}\, q(\mathbf{S}) \log \frac{p(\mathbf{X}, \mathbf{S}|\mathbf{T}, \mathbf{V})}{q(\mathbf{S})} \equiv \mathcal{B}[q(\mathbf{S})]\end{aligned}$$

by Jensens' inequality [Bishop, 2006], defining a lower bound on the log likelihood. Here, $q(\mathbf{S})$ is an instrumental distribution over the set of latent sources, with the property that $q(\mathbf{S}) = 0$ if and only if $p(\mathbf{X}, \mathbf{S}|\mathbf{T}, \mathbf{V}) = 0$. The lower bound is tight when the instrumental distribution is the posterior of the latent sources:

$$\arg\max_{q(\mathbf{S})} \mathcal{B}[q(\mathbf{S})] = p(\mathbf{S}|\mathbf{X}, \mathbf{T}, \mathbf{V})$$

Hence we can use an iterative coordinate ascent scheme to maximize the log likelihood. The first step, called the expectation (E) step, is to compute the posterior distribution, which we will see has the form of a multivariate normal. Because this is an exponential family, we only need to compute the sufficient statistics, which is why we call this the expectation step. The second step is called the maximization (M) step because we find the maximum likelihood $\mathbf{T}$ and $\mathbf{V}$ holding $q(\mathbf{S})$ fixed. The two steps of the expectation maximization algorithm (EM) are summarized as:

$$\begin{aligned}&\text{E-step} && q(\mathbf{S})^{(n)} = p(\mathbf{S}|\mathbf{X}, \mathbf{T}^{(n-1)}\mathbf{V}^{(n-1)}) \\ &\text{M-step} && \left\{\mathbf{T}^{(n)}, \mathbf{V}^{(n)}\right\} = \arg\max_{\mathbf{T},\mathbf{V}} \langle \log p(\mathbf{S}, \mathbf{X}|\mathbf{T}, \mathbf{V})\rangle_{q(\mathbf{S})}\end{aligned}$$

7.2.2 Expectation Step

In this section we derive the posterior of the latent sources

$$p(\mathbf{S}|\mathbf{X}, \mathbf{T}, \mathbf{V}) = \frac{p(\mathbf{S}, \mathbf{X}|\mathbf{T}, \mathbf{V})}{p(\mathbf{X}|\mathbf{T}, \mathbf{V})}$$

The terms in the expression for the log probability density of the posterior are given by

$$\log p(\mathbf{S}|\mathbf{X}, \mathbf{T}, \mathbf{V}) = \sum_{\nu,\tau} \left(\sum_i \left(-\frac{D}{2} \frac{|s_{\nu,i,\tau}|^2}{t_{\nu,i} v_{i,\tau}} \right) + \frac{D}{2} \frac{|\sum_i s_{\nu,i,\tau}|^2}{\sum_i t_{\nu,i} v_{i,\tau}} \right) + \cdots \tag{7.5}$$

This defines a multivariate normal distribution over the latent sources, for which we will adopt the following notation: $s_{\nu,\tau} = [s_{\nu,1,\tau}, \ldots, s_{\nu,I,\tau}]$, 1 is a I element row vector of ones, so that we can write $\sum_i s_{\nu,i,\tau} = 1 s_{\nu,\tau}$.

Let $A_{\nu,\tau}$ be a $I \times I$ diagonal (covariance) matrix with ith element $t_{\nu,i}v_{i,\tau}$. The above expression is rewritten as

$$\log p(\mathbf{S}|\mathbf{X},\mathbf{T},\mathbf{V}) = \sum_{\nu,\tau}\left(-\frac{D}{2}\mathrm{Tr}\, A_{\nu,\tau}^{-1}s_{\nu,\tau}s_{\nu,\tau}^H + \frac{D}{2}\mathrm{Tr}\,\frac{1^\top 1 s_{\nu,\tau}s_{\nu,\tau}^H}{1A_{\nu,\tau}1^\top}\right) + \cdots$$

which, after some manipulations (see Section B.2), becomes

$$= \sum_{\nu,\tau} -\frac{D}{2}\mathrm{Tr}\left(s_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top 1 s_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right)^H \left(A_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top 1 A_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right)\left(s_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top 1 s_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right) + \cdots$$

$$= \sum_{\nu,\tau} -\frac{D}{2}\mathrm{Tr}\left(s_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top x_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right)^H \left(A_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top 1 A_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right)\left(s_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top x_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right) + \cdots$$

This is a multivariate normal distribution, as we can write

$$p(\mathbf{S}|\mathbf{X},\mathbf{T},\mathbf{V}) = \prod_{\nu,\tau}\mathcal{N}\left(s_{\nu,\tau}; \frac{A_{\nu,\tau}1^\top x_{\nu,\tau}}{1A_{\nu,\tau}1^\top}, A_{\nu,\tau} - \frac{A_{\nu,\tau}1^\top 1 A_{\nu,\tau}}{1A_{\nu,\tau}1^\top}\right) = \prod_{\nu,\tau}\mathcal{N}(s_{\nu,\tau};\mu_{\nu,\tau},\Sigma_{\nu,\tau})$$

and the standard results for the sufficient statistics of the posterior are:

$$\begin{aligned}\langle s_{\nu,\tau}\rangle &= \mu_{\nu,\tau} \\ \left\langle s_{\nu,\tau}s_{\nu,\tau}^H\right\rangle &= \mu_{\nu,\tau}\mu_{\nu,\tau}^H + \Sigma_{\nu,\tau}\end{aligned}$$

By defining a positive quantity called the *responsibility* by Cemgil and Dikmen [2007]

$$\kappa_{\nu,i,\tau} = \frac{t_{\nu,i}v_{i,\tau}}{\sum_{i'} t_{\nu,i'}v_{i',\tau}}$$

we can write the correlations as

$$\left\langle |s_{\nu,i,\tau}|^2\right\rangle = Dt_{\nu,i}v_{i,\tau}\left(1-\kappa_{\nu,i,\tau}\right) + \kappa_{\nu,i,\tau}^2|x_{\nu,\tau}|^2$$

7.2.3 Maximization Step

In this section, we will present the M step as a coordinate ascent in $\mathbf{T}$ and $\mathbf{V}$. Other schemes such as gradient descent and Hessian based approaches are possible (see for example Dhillon and Sra [2006]), however they involve more computation and storage requirements. The update rule for the templates is given by

$$\frac{\partial}{\partial t_{\nu,i}}\left\langle \log p\left(\mathbf{X},\mathbf{S}|\mathbf{T},\mathbf{V}\right)\right\rangle = \frac{D}{2}\sum_{\tau}\left(\frac{|s_{\nu,i,\tau}|^2}{t_{\nu,i}^2 v_{i,\tau}} - \frac{1}{t_{\nu,i}}\right) = 0$$

$$t_{\nu,i} = \frac{1}{T}\sum_{\tau}\frac{|s_{\nu,i,\tau}|^2}{v_{i,\tau}}$$

and the update rule for the excitations is given by:

$$\frac{\partial}{\partial v_{i,\tau}} \langle \log p\left(\mathbf{X}, \mathbf{S} | \mathbf{T}, \mathbf{V}\right) \rangle = \frac{D}{2} \sum_{\nu} \left(\frac{|s_{\nu,i,\tau}|^2}{t_{\nu,i} v_{i,\tau}^2} - \frac{1}{v_{i,\tau}} \right) = 0$$

$$v_{i,\tau} = \frac{1}{F} \sum_{\nu} \frac{|s_{\nu,i,\tau}|^2}{t_{\nu,i}}$$

The summations in the update rules can be carried out by means of efficient matrix multiplications. Note that it is unnecessary (and also expensive) to calculate the complete sufficient statistics of the posterior over the latent sources. All that is required is the summations over frequency and time of the individual correlations (the diagonal elements of the covariance matrix).

7.3 Bayesian Hierarchical Model

To exploit the power of Bayesian inference, we place conjugate priors on the templates and excitations in the following hierarchical model.

$$\begin{aligned} t_{\nu,i} &\sim \mathcal{IG}(t_{\nu,i}; a^t_{\nu,i}, b^t_{\nu,i} a^t_{\nu,i}) \\ v_{i,\tau} &\sim \mathcal{IG}(v_{i,\tau}; a^v_{i,\tau}, b^v_{i,\tau} a^v_{i,\tau}) \\ s_{\nu,i,\tau} &\sim \mathcal{N}(s_{\nu,i,\tau}; 0, t_{\nu,i} v_{i,\tau}) \\ x_{\nu,\tau} &= \sum_i s_{\nu,i,\tau} \end{aligned} \tag{7.6}$$

The inverse-gamma distribution (Figure 7.2 on page 106) is a conjugate prior to the variance of the normal distribution. This particular parametrization has the following interpretation: $\langle 1/t_{\nu,i} \rangle = 1/b^t_{\nu,i}$ and $\langle 1/v_{i,\tau} \rangle = 1/b^v_{i,\tau}$ under the prior, so the scale parameters approximately gives the expected values of the templates and excitations. The standard deviation is given by $\frac{a}{(a-1)\sqrt{a-2}}$ which decreases with a, hence the scale parameter can represent the sparsity of the representation. A high value of a means a low standard deviation from the scale parameter, hence most of the coefficients have similar magnitudes, implying a full representation. A low value of a means a high standard deviation from the scale parameter, hence most of the coefficients of the representation will be close to zero as shown in Figure 7.2 on page 106, favouring a spare representation.

The joint probability distribution of this model is given by

$$p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V}) = p(\mathbf{X}, \mathbf{S} | \mathbf{T}, \mathbf{V}) p(\mathbf{T}) p(\mathbf{V})$$

from which we can consider a number of inference tasks. These typically involve calculating the posterior $p(\mathbf{S}, \mathbf{T}, \mathbf{V} | \mathbf{X})$ and the marginal likelihood (also known as the evidence) given the hyperparameters $p(\mathbf{X})$.

7.3.1 Inference by Variational Bayes

The development of the Variational Bayes inference algorithm [Bishop, 2006, Ghahramani and Beal, 2001] is similar to the EM algorithm. Again we approximate the log marginal likelihood by means of an instrumental

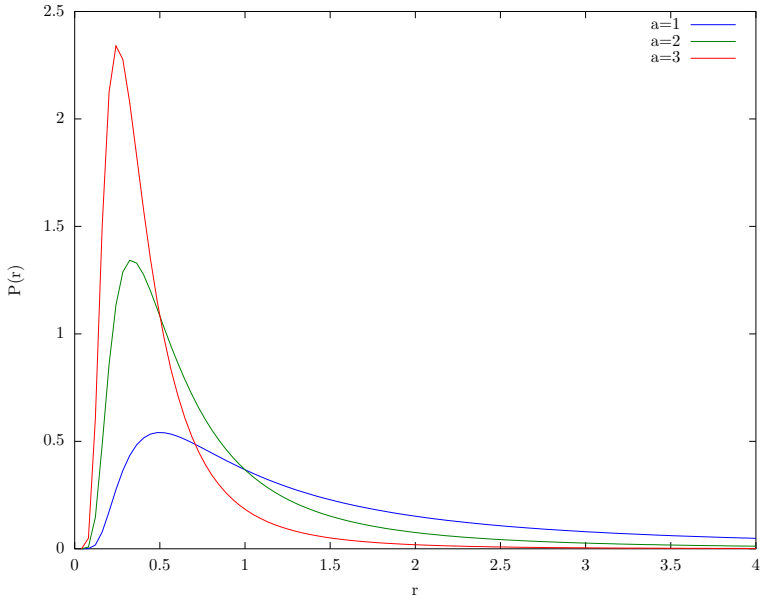


Figure 7.2: The inverse-gamma distribution, $p(r) = \mathcal{IG}(r; a, a)$ for different a, and scale parameter $b = 1$

distribution:

$$\begin{aligned}
\mathcal{L}_{\mathbf{X}} \equiv \log p(\mathbf{X}) &= \log \int \mathrm{d}\mathbf{S}\mathrm{d}\mathbf{T}\mathrm{d}\mathbf{V}\, p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V}) \\
&= \log \int \mathrm{d}\mathbf{S}\mathrm{d}\mathbf{T}\mathrm{d}\mathbf{V}\, \frac{q(\mathbf{S}, \mathbf{T}, \mathbf{V})}{q(\mathbf{S}, \mathbf{T}, \mathbf{V})} p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V}) \\
&\geq \int \mathrm{d}\mathbf{S}\mathrm{d}\mathbf{T}\mathrm{d}\mathbf{V}\, q(\mathbf{S}, \mathbf{T}, \mathbf{V}) \log \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V})}{q(\mathbf{S}, \mathbf{T}, \mathbf{V})} \equiv \mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})]
\end{aligned}$$

The bound is tight when the instrumental distribution is equal to the posterior:

$$q(\mathbf{S}, \mathbf{T}, \mathbf{V}) = p(\mathbf{S}, \mathbf{T}, \mathbf{V}|\mathbf{X})$$

However the posterior distribution is intractable, so instead we assume a factorized form

$$q(\mathbf{S}, \mathbf{T}, \mathbf{V}) = q(\mathbf{S})q(\mathbf{T})q(\mathbf{V}) = \left(\prod_{\nu,\tau} q(s_{\nu,\tau})\right)\left(\prod_{\nu,i} q(t_{\nu,i})\right)\left(\prod_{i,\tau} q(v_{i,\tau})\right)$$

This particular approximation is known as the mean field approximation. It can be shown that updating the sufficient statistics of $q(\mathbf{S})$, $q(\mathbf{T})$ or $q(\mathbf{V})$, holding both of the other distributions constant, leads to a monotonically increasing bound with each iteration $\mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})^{(n+1)}] \geq \mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})^{(n)}]$. To illustrate the similarity between VB and EM, we will choose the following ordering of update rules.

The approximate E-step is:

$$q(\mathbf{S})^{(n)} \propto \exp\left(\langle \log p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V})\rangle_{q(\mathbf{T})^{(n-1)}q(\mathbf{V})^{(n-1)}}\right)$$

then the approximate M-step involves iterating

$$q(\mathbf{T})^{(n+k)} \propto \exp\left(\langle \log p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V})\rangle_{q(\mathbf{S})^{(n)}q(\mathbf{V})^{(n+k-1)}}\right)$$
$$q(\mathbf{V})^{(n+k)} \propto \exp\left(\langle \log p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V})\rangle_{q(\mathbf{S})^{(n)}q(\mathbf{T})^{(n+k)}}\right)$$

for $k = 1, \ldots, K$ until convergence as defined in 7.3.1.2.

An alternative means of Bayesian model selection for the Gaussian variance model with half-normal priors on the factor matrices is investigated by Févotte [2010] using a model selection criteria. In Févotte and Cemgil [2009] Bayesian model selection using both Variational Bayes and MCMC is sketched for a number of NMF models, including the Gaussian variance model.

7.3.1.1 Variational update equations and sufficient statistics

The update rule for the expectation step follows from (7.5):

$$q(s_{\nu,\tau}) \propto \exp\left(\sum_i \left(-\frac{D}{2}\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle |s_{\nu,i,\tau}|^2\right) + \frac{D}{2}\frac{|\sum_i s_{\nu,i,\tau}|^2}{\sum_i \left(\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle\right)^{-1}}\right)$$

The calculation is the same as for the E-step of the EM algorithm, but the covariance matrix $A_{\nu,\tau}$ has elements $\left(\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle\right)^{-1}$ along the diagonal. The responsibilities are

$$\kappa_{\nu,i,\tau} = \frac{\left(\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle\right)}{\sum_i \left(\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle\right)}$$

and the correlations that we need for the M-step are

$$\left\langle |s_{\nu,i,\tau}|^2 \right\rangle = \left(\left\langle t_{\nu,i}^{-1}\right\rangle \left\langle v_{i,\tau}^{-1}\right\rangle\right)^{-1}(1 - \kappa_{\nu,i,\tau}) + \kappa_{\nu,i,\tau}^2 |x_{\nu,\tau}|^2 \tag{7.7}$$

The update equations and sufficient statistics for the templates and excitations follow from the properties of the inverse-gamma distribution:

$$q(t_{\nu,i}) \propto \exp\left(-(a^t_{\nu,i} + \tfrac{DT}{2} + 1)\log t_{\nu,i} - (a^t_{\nu,i} b^t_{\nu,i} + \tfrac{D}{2}\sum_\tau \left\langle |s_{\nu,i,\tau}|^2 \right\rangle \left\langle v^{-1}_{i,\tau} \right\rangle) \left\langle t^{-1}_{\nu,i} \right\rangle\right) \propto \mathcal{IG}(t_{\nu,i}; \alpha^t_{\nu,i}, \beta^t_{\nu,i})$$

$$\alpha^t_{\nu,i} = a^t_{\nu,i} + \tfrac{DT}{2} \qquad \beta^t_{\nu,i} = a^t_{\nu,i} b^t_{\nu,i} + \tfrac{D}{2}\sum_\tau \left\langle |s_{\nu,i,\tau}|^2 \right\rangle \left\langle v^{-1}_{i,\tau} \right\rangle$$

$$\left\langle t^{-1}_{\nu,i} \right\rangle = \frac{\alpha^t_{\nu,i}}{\beta^t_{\nu,i}} \qquad \langle \log t_{\nu,i} \rangle = -\Psi(\alpha^t_{\nu,i}) + \log \beta^t_{\nu,i} \tag{7.8}$$

$$q(v_{i,\tau}) \propto \exp\left(-(a^v_{i,\tau} + \tfrac{DF}{2} + 1)\log v_{i,\tau} - (a^v_{i,\tau} b^v_{i,\tau} + \tfrac{D}{2}\sum_\nu \left\langle |s_{\nu,i,\tau}|^2 \right\rangle \left\langle t^{-1}_{\nu,i} \right\rangle) \left\langle v^{-1}_{i,\tau} \right\rangle\right) \propto \mathcal{IG}(v_{i,\tau}; \alpha^v_{i,\tau}, \beta^v_{i,\tau})$$

$$\alpha^v_{i,\tau} = a^v_{i,\tau} + \tfrac{DF}{2} \qquad \beta^v_{i,\tau} = a^v_{i,\tau} b^v_{i,\tau} + \tfrac{D}{2}\sum_\nu \left\langle |s_{\nu,i,\tau}|^2 \right\rangle \left\langle t^{-1}_{\nu,i} \right\rangle$$

$$\left\langle v^{-1}_{i,\tau} \right\rangle = \frac{\alpha^v_{i,\tau}}{\beta^v_{i,\tau}} \qquad \langle \log v_{i,\tau} \rangle = -\Psi(\alpha^v_{i,\tau}) + \log \beta^v_{i,\tau} \tag{7.9}$$

We retain the attractiveness of being able to perform these update equations as matrix operations. Note that expensive evaluations of the digamma function $\Psi(\alpha)$ can be precomputed, as the posterior shape parameters are constant.

7.3.1.2 The Variational Bound

The variational bound is a lower bound on the marginal log likelihood and also can be used to define convergence for the E and M steps.

$$\mathcal{B}[q(\mathbf{S},\mathbf{T},\mathbf{V})] = \langle \log p(\mathbf{X},\mathbf{S},\mathbf{T},\mathbf{V}) \rangle_q + H[q(\mathbf{S},\mathbf{T},\mathbf{V})]$$

$H[q(\mathbf{S},\mathbf{T},\mathbf{V})]$ denotes the entropy of the variational distribution q, which is defined as $-\langle \log q(\mathbf{S},\mathbf{T},\mathbf{V}) \rangle_q$.

The following is the complete expression for the variational bound immediately after the E-step. The first two rows are the combined energy and entropy of the latent sources.

$$\begin{aligned}
\mathcal{B}[q(\mathbf{S},\mathbf{T},\mathbf{V})] = &+ \sum_{\nu,\tau}\left(-\frac{D}{2}\left(\sum_i \left\langle t^{-1}_{\nu,i}\right\rangle^{-1}\left\langle v^{-1}_{i,\tau}\right\rangle^{-1}\right)^{-1}|x_{\nu,\tau}|^2 - \frac{D}{2}\log\frac{2\pi}{D} + \frac{D}{2}\log\left(\sum_i \left\langle t^{-1}_{\nu,i}\right\rangle^{-1}\left\langle v^{-1}_{i,\tau}\right\rangle^{-1}\right)\right) \\
&- \frac{D}{2}\sum_{\nu,i,\tau}\left(-\langle \log t_{\nu,i}\rangle - \log\left\langle t^{-1}_{\nu,i}\right\rangle - \langle \log v_{i,\tau}\rangle - \log\left\langle v^{-1}_{i,\tau}\right\rangle\right) \qquad \mathcal{F}[\mathbf{S}] + H[q(\mathbf{S})] \\
&+ \sum_{\nu,i}\left(-(a^t_{\nu,i}+1)\langle \log t_{\nu,i}\rangle - a^t_{\nu,i}b^t_{\nu,i}\left\langle t^{-1}_{\nu,i}\right\rangle + a^t_{\nu,i}\log(a^t_{\nu,i}b^t_{\nu,i}) - \log\Gamma(a^t_{\nu,i})\right) \qquad \mathcal{F}[\mathbf{T}] \\
&- \sum_{\nu,i}\left(-(\alpha^t_{\nu,i}+1)\langle \log t_{\nu,i}\rangle - \beta^t_{\nu,i}\left\langle t^{-1}_{\nu,i}\right\rangle + \alpha^t_{\nu,i}\log\beta^t_{\nu,i} - \log\Gamma(\alpha^t_{\nu,i})\right) \qquad H[q(\mathbf{T})] \\
&+ \sum_{\tau,i}\left(-(a^v_{i,\tau}+1)\langle \log v_{i,\tau}\rangle - a^v_{i,\tau}b^v_{i,\tau}\left\langle v^{-1}_{i,\tau}\right\rangle + a^v_{i,\tau}\log(a^v_{i,\tau}b^v_{i,\tau}) - \log\Gamma(a^v_{i,\tau})\right) \qquad \mathcal{F}[\mathbf{V}] \\
&- \sum_{\tau,i}\left(-(\alpha^v_{i,\tau}+1)\langle \log v_{i,\tau}\rangle - \beta^v_{i,\tau}\left\langle v^{-1}_{i,\tau}\right\rangle + \alpha^v_{i,\tau}\log\beta^v_{i,\tau} - \log\Gamma(\alpha^v_{i,\tau})\right) \qquad H[q(\mathbf{V})]
\end{aligned} \tag{7.10}$$

The 'energy' notation used to label the summations here is:

$$\mathcal{F}[\mathbf{S}] = \langle \log p(\mathbf{X}, \mathbf{S}|\mathbf{T}, \mathbf{V}) \rangle_q$$
$$\mathcal{F}[\mathbf{T}] = \langle \log p(\mathbf{T}) \rangle_q$$
$$\mathcal{F}[\mathbf{V}] = \langle \log p(\mathbf{V}) \rangle_q$$

The calculation of the variational bound can be implemented efficiently using matrix operations, just as with the update equations. Expensive evaluations of $\log \Gamma(\alpha)$ can be precomputed.

Once the templates or excitations are updated in the M-step, the variational bound cannot be derived in closed form. However, during the M-step, $H[q(\mathbf{S})]$ remains constant, so we do not need to take it account when determining whether the M-step iterations have converged. This is convenient because $H[q(\mathbf{S})]$ is not straightforward to derive in isolation from $\mathcal{F}[\mathbf{S}]$. We therefore need to confirm after each update in the M-step that the quantity

$$\mathcal{F}[\mathbf{S}] + \mathcal{F}[\mathbf{T}] + H[q(\mathbf{T})] + \mathcal{F}[\mathbf{V}] + H[q(\mathbf{V})] \tag{7.11}$$

increases, otherwise we perform another E-step at this stage (see Algorithm 7.1). The values for $\mathcal{F}[\mathbf{T}]$, $H[q(\mathbf{T})]$, $\mathcal{F}[\mathbf{V}]$ and $H[q(\mathbf{V})]$ are as in (7.10), however $\mathcal{F}[\mathbf{S}]$ during the M-step is given by

$$\mathcal{F}[\mathbf{S}] = \sum_{\nu} \sum_{\tau} \sum_{i} \left(-\frac{D}{2} \left\langle t_{\nu,i}^{-1} \right\rangle \left\langle v_{i,\tau}^{-1} \right\rangle \left\langle s_{\nu,i,\tau}^{2} \right\rangle - \frac{D}{2} \log \frac{2\pi}{D} - \frac{D}{2} \left\langle \log t_{\nu,i} \right\rangle - \frac{D}{2} \left\langle \log v_{i,\tau} \right\rangle \right)$$

where $\left\langle s_{\nu,i,\tau}^{2} \right\rangle$ is given by (7.7). $\mathcal{F}[\mathbf{S}]$ may be calculated efficiently using matrix update equations from the sufficient statistics of $\mathbf{T}$ and $\mathbf{V}$ immediately prior to being updated during the M-step.

7.3.1.3 Hyperparameter Optimization

Hyperparameter optimization involves maximizing the variational bound with respect to the hyperparameters. The components of the variational bound that correspond to maximizing the bound are $\mathcal{F}[\mathbf{T}]$ for the template hyperparameters, and $\mathcal{F}[\mathbf{V}]$ for the excitation hyperparameters[1]. The resulting expressions for optimization are very similar to expressions for finding maximum likelihood estimates of the parameters of an inverse-gamma distribution.

Hyperparameter optimization is used for training the priors of the template and excitation matrices using labelled data. The hyperparameters will be tied over some subset of the elements of the template or excitation factorization matrices. For example, we typically do not *a priori* know the length of time over which we will observe data for the model. For the model to be identifiable, we are forced to tie the excitation hyperparameters across the rows of the excitation matrix.

Because of the numerous possibilities for how we set up the optimization, we will only outline the process here for a single set of parameters $\mathcal{M}$ with variances $\{r_m\}$ over which either the shape parameter or the scale parameter is tied. This allows the shape parameters and the scale parameters to be tied over different subsets of the variances, for example we might want to have a global shape parameter for all of the template variances, but have a scale parameter for each column of the template matrix.

[1] In (7.10) the entropy expression $H[q(\mathbf{T})]$ is not directly dependent on the values of the hyperparameters, but is dependent only through the variational distribution $q(\mathbf{T})$, which is updated during the M-Step for the templates. The same is true for the excitation hyperparameters. Therefore the entropy expressions are not used to optimize the hyperparameters in this section.

To optimize a single scale parameter $b_{\mathcal{M}}$ which is tied over the variances $\{r_m\}$ with corresponding shape parameters $\{a_m\}$, we maximize the following expression

$$\mathcal{B}\left(b_{\mathcal{M}}\right) = \sum_{m \in \mathcal{M}} \left(-\left(a_m + 1\right) \left\langle \log r_m \right\rangle - a_m b_{\mathcal{M}} \left\langle r_m^{-1} \right\rangle + a_m \log(a_m b_{\mathcal{M}}) - \log \Gamma(a_m)\right)$$

by setting the derivative to zero, i.e.

$$\frac{\partial \mathcal{B}}{\partial b_{\mathcal{M}}} = \sum_{m \in \mathcal{M}} \left(-a_m \left\langle r_m^{-1} \right\rangle + \frac{a_m}{b_{\mathcal{M}}}\right) = 0 \tag{7.12}$$

giving an update rule:

$$b_{\mathcal{M}} \leftarrow \frac{\sum_{\mathcal{M}} a_m}{\sum_{\mathcal{M}} a_m \left\langle r_m^{-1} \right\rangle} \tag{7.13}$$

To optimize a single shape parameter $a_{\mathcal{M}}$ which is tied over the variances $\{r_m\}$ with corresponding scale parameters $\{b_m\}$, we maximize the following expression

$$\mathcal{B}\left(a_{\mathcal{M}}\right) = \sum_{m \in \mathcal{M}} \left(-\left(a_{\mathcal{M}} + 1\right) \left\langle \log r_m \right\rangle - a_{\mathcal{M}} b_m \left\langle r_m^{-1} \right\rangle + a_{\mathcal{M}} \log(a_{\mathcal{M}} b_m) - \log \Gamma(a_{\mathcal{M}})\right)$$

by setting the derivative to zero, i.e.

$$\frac{\partial \mathcal{B}}{\partial a} = \sum_{m \in \mathcal{M}} \left(- \left\langle \log r_m \right\rangle - b_m \left\langle r_m^{-1} \right\rangle + \log a + 1 + \log b_m - \Psi(a)\right)$$

giving the following expression:

$$\log a_{\mathcal{M}} - \Psi(a_{\mathcal{M}}) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(\left\langle \log r_m \right\rangle + b_m \left\langle r_m^{-1} \right\rangle - 1 - \log b_m\right)$$

Equations of this form appear in the maximum likelihood estimate of Gamma distributions. a can be found by Newton iterations. Denote the right hand side as c, then the following is the Newton-Raphson update:

$$a_{\mathcal{M}} \leftarrow a_{\mathcal{M}} - \frac{\log a_{\mathcal{M}} - \Psi(a_{\mathcal{M}}) - c}{1/a_{\mathcal{M}} - \Psi'(a_{\mathcal{M}})} \tag{7.14}$$

Hyperparameter optimization is performed after the variational bound no longer increases through the E-Step and M-Step updates. The entire structure of the algorithm is described in Algorithm 7.1.

Algorithm 7.1 Variational Bayes for the Gaussian variance model, with hyperparameter optimization

- Sample point estimates $t_{\nu,i}$ and $v_{i,\tau}$ from the priors (7.6)
- Initialize sufficient statistics $\left\langle t_{\nu,i}^{-1} \right\rangle \leftarrow (t_{\nu,i})^{-1}, \langle \log t_{\nu,i} \rangle \leftarrow \log t_{\nu,i}, \left\langle v_{i,\tau}^{-1} \right\rangle \leftarrow (v_{i,\tau})^{-1}, \langle \log v_{i,\tau} \rangle \leftarrow \log v_{i,\tau}$ from the point estimates
- Calculate the variational bound $\mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})]^{(0)}$ from (7.10)
- for $n = 1, \ldots$
  - for $k = 1 \ldots$
    - Update the template sufficient statistics $\left\langle t_{\nu,i}^{-1} \right\rangle, \langle \log t_{\nu,i} \rangle$ (7.8)
    - Update the excitation sufficient statistics $\left\langle v_{i,\tau}^{-1} \right\rangle, \langle \log v_{i,\tau} \rangle$ (7.9)
    - If the quantity (7.11) increases, then continue, otherwise break
  - Calculate the variational bound $\mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})]^{(n)}$
  - If $\mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})]^{(n)} = \mathcal{B}[q(\mathbf{S}, \mathbf{T}, \mathbf{V})]^{(n-1)}$ then
    - update each tied shape parameter using (7.14)
    - update each tied scale parameter using (7.13)
    - If the updates result in no changes to the hyperparameters, exit the algorithm

7.3.2 Markov Chain Monte-Carlo

7.3.2.1 Gibbs Sampler

A possible Gibbs sampler for the model involves sampling the blocks:

$$\begin{aligned}
\mathbf{S}^{(n+1)} &\sim p(\mathbf{S}|\mathbf{X}, \mathbf{T}^{(n)}, \mathbf{V}^{(n)}) \\
\mathbf{T}^{(n+1)} &\sim p(\mathbf{T}|\mathbf{S}^{(n+1)}, \mathbf{V}^{(n)}) \\
\mathbf{V}^{(n+1)} &\sim p(\mathbf{V}|\mathbf{S}^{(n+1)}, \mathbf{T}^{(n+1)})
\end{aligned}$$

The marginal likelihood is estimated by Chib's method [Chib, 1995] around the mode $\{\mathbf{S}^*, \mathbf{T}^*, \mathbf{V}^*\}$ found by running the Gibbs sampler. The marginal likelihood is decomposed as:

$$\log p(\mathbf{X}) = \log p(\mathbf{X}, \mathbf{S}^*, \mathbf{T}^*, \mathbf{V}^*) - \log p(\mathbf{S}^*|\mathbf{X}) - \log p(\mathbf{T}^*|\mathbf{S}^*) - \log p(\mathbf{V}^*|\mathbf{S}^*, \mathbf{T}^*)$$

The second term is found by the Monte-Carlo estimate:

$$p(\mathbf{S}^*|\mathbf{X}) \approx \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{S}^*|\mathbf{X}, \mathbf{T}^{(n)}, \mathbf{V}^{(n)})$$

with $\{\mathbf{T}^{(n)}, \mathbf{V}^{(n)})\}$ returned by the Gibbs' sampler. The third term is found by the Monte-Carlo estimate:

$$p(\mathbf{T}^*|\mathbf{S}^*) \approx \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{T}^*|\mathbf{S}^*, \mathbf{V}^{(m)})$$

requiring a further M samples from the reduced Gibbs sampler, clamping $\mathbf{S} = \mathbf{S}^*$

$$\mathbf{T}^{(m+1)} \sim p(\mathbf{T}|\mathbf{S}^*, \mathbf{V}^{(m)})$$
$$\mathbf{V}^{(m+1)} \sim p(\mathbf{V}|\mathbf{S}^*, \mathbf{T}^{(m+1)})$$

Sampling Latent Sources The distribution $p(s_{\nu,1:I,\tau}|x_{\nu,\tau}, t_{\nu,1:I}, v_{1:I,\tau})$ is a degenerate multivariate normal distribution, because $p(x_{\nu,\tau}|s_{\nu,1:I,\tau})$ is itself degenerate. As the covariance matrix is not positive definite, we cannot form the Cholesky factor and therefore sample from this distribution directly. Instead, we sample from the reduced distribution $p(s_{\nu,2:I,\tau}|x_{\nu,\tau}, t_{\nu,2:I}, v_{2:I,\tau})$ and then set $s_{\nu,1,\tau} = x_{\nu,\tau} - \sum_{i=2}^{I} s_{\nu,i,\tau}$. In the rest of this section, we drop the subscripts ν, τ for brevity. First observe that

$$p(x|s_{2:I}) = \mathcal{N}(x; 1s_{2:I}, t_1 v_1)$$

where 1 is an $(I-1)$ row vector of ones. It follows that the posterior is

$$\begin{aligned}\log p(s_{2:I}|x, t_{2:I} v_{2:I}) &= -\frac{D}{2}\frac{|x - 1s_{2:I}|^2}{t_1 v_1} - \frac{D}{2} s_{2:I}^H A^{-1} s_{2:I} + \dots \\ &= \frac{D}{t_1 v_1} s_{2:I}^H 1^\top x - \frac{D}{2}\mathrm{Tr}\left(\frac{1}{t_1 v_1} 1^\top 1 + A^{-1}\right) s_{2:I} s_{2:I}^H\end{aligned}$$

where A is a diagonal matrix with elements $t_i v_i, i = 2, \dots, I$. The posterior is therefore a multivariate normal with covariance matrix and mean

$$\begin{aligned}\Sigma &= \left(\frac{1}{t_1 v_1} 1^\top 1 + A^{-1}\right)^{-1} = A - \frac{A 1^\top 1 A}{t_1 v_1 + 1 A 1^\top} \\ \mu &= \frac{1}{t_1 v_1}\Sigma 1^\top x\end{aligned}$$

Note that the covariance matrix is formed by downdating the diagonal matrix A with the vector $1A$ scaled by $(t_1 v_1 + 1A1^\top)^{-1}$. The Cholesky factorization of the covariance matrix can be computed more efficiently by downdating the Cholesky factor of A than by calculating the full factorization. The Cholesky factor of A is a diagonal matrix with elements $\sqrt{t_i v_i}, i = 2, \dots, I$.

Single Source The following is a description of the Gibbs sampler for the single source case, where the source is directly observed: $\mathbf{S} = \mathbf{X}$. We mention this as a special case as the previous expressions for estimating the marginal likelihood using Chib's method do not apply here. The algorithm iterates

$$\mathbf{T}^{(n+1)} \sim p(\mathbf{T}|\mathbf{X}, \mathbf{V}^{(n)})$$
$$\mathbf{V}^{(n+1)} \sim p(\mathbf{V}|\mathbf{X}, \mathbf{T}^{(n+1)})$$

and the marginal likelihood at the mode $\{\mathbf{T}^*, \mathbf{V}^*\}$ found from the above run is:

$$\log p(\mathbf{X}) = \log p(\mathbf{X}, \mathbf{T}^*, \mathbf{V}^*) - \log p(\mathbf{T}^*|\mathbf{X}) - \log p(\mathbf{V}^*|\mathbf{X}, \mathbf{T}^*)$$

and the second term is found by the Monte-Carlo estimate

$$p(\mathbf{T}^*|\mathbf{X}) \approx \frac{1}{N}\sum_{n=1}^{N} p(\mathbf{T}^*|\mathbf{X}, \mathbf{V}^{(n)})$$

7.3.2.2 Metropolis-Hastings

The particular scheme we propose here marginalizes the latent sources, and attempts to draw samples from the posterior $p(\mathbf{T}, \mathbf{V}|\mathbf{X})$ directly by constructing a Markov chain which draws samples from $p(\mathbf{T}|\mathbf{X}, \mathbf{V})$ and $p(\mathbf{V}|\mathbf{X}, \mathbf{T})$ in sequence. However, these distributions cannot be sampled directly, so we resort to a Metropolis-Hastings (MH) algorithm. Note that the posterior can be written as:

$$p(\mathbf{T}, \mathbf{V}|\mathbf{X}) = \frac{1}{p(\mathbf{X})} p(\mathbf{X}|\mathbf{T}, \mathbf{V}) \propto p(\mathbf{X}|\mathbf{T}, \mathbf{V}) p(\mathbf{T}) p(\mathbf{V})$$

$p(\mathbf{X}|\mathbf{T}, \mathbf{V})$ is already given in (7.2). The MH algorithm requires a proposal density with the same coverage as the posterior distribution being sampled. We suggest using the inverse-gamma distributions $q(\mathbf{T})$ and $q(\mathbf{V})$ in (7.8) and (7.9) as proposal distributions for the template and excitation parameters respectively, substituting the sufficient statistics with the current point estimates. These are suitable for inferring the posterior $p(\mathbf{T}, \mathbf{V}|\mathbf{X})$ as the method used to derive them also involved marginalizing the latent sources (the E-Step of the Variational Bayes algorithm). For the MH algorithm we denote these proposal densities as $q(\mathbf{T}, \mathbf{T}'|\mathbf{X}, \mathbf{V})$, the probability of moving from $\mathbf{T}$ to $\mathbf{T}'$. The MH algorithm simply involves iterating between the moves

$$\begin{aligned}
\mathbf{T}' &\sim q(\mathbf{T}^{(n)}, \mathbf{T}'|\mathbf{X}, \mathbf{V}^{(n)}) \\
\mathbf{T}^{(n+1)} &= \begin{cases} \mathbf{T}' & \text{if } \alpha(\mathbf{T}^{(n)}, \mathbf{T}'|\mathbf{X}, \mathbf{V}^{(n)}) \leq \mathcal{U}(0,1) \\ \mathbf{T}^{(n)} & \text{otherwise} \end{cases} \\
\mathbf{V}' &\sim q(\mathbf{V}^{(n)}, \mathbf{V}'|\mathbf{X}, \mathbf{T}^{(n+1)}) \\
\mathbf{V}^{(n+1)} &= \begin{cases} \mathbf{V}' & \text{if } \alpha(\mathbf{V}^{(n)}, \mathbf{V}'|\mathbf{X}, \mathbf{T}^{(n+1)}) \leq \mathcal{U}(0,1) \\ \mathbf{V}^{(n)} & \text{otherwise} \end{cases}
\end{aligned}$$

where the acceptance ratios are given by

$$\begin{aligned}
\alpha(\mathbf{T}, \mathbf{T}'|\mathbf{X}, \mathbf{V}) &= \min\left\{1, \frac{p(\mathbf{X}, \mathbf{T}', \mathbf{V})}{p(\mathbf{X}, \mathbf{T}, \mathbf{V})} \frac{q(\mathbf{T}', \mathbf{T}|\mathbf{X}, \mathbf{V})}{q(\mathbf{T}, \mathbf{T}'|\mathbf{X}, \mathbf{V})}\right\} \\
\alpha(\mathbf{V}, \mathbf{V}'|\mathbf{X}, \mathbf{T}) &= \min\left\{1, \frac{p(\mathbf{X}, \mathbf{T}, \mathbf{V}')}{p(\mathbf{X}, \mathbf{T}, \mathbf{V})} \frac{q(\mathbf{V}', \mathbf{V}|\mathbf{X}, \mathbf{T})}{q(\mathbf{V}, \mathbf{V}'|\mathbf{X}, \mathbf{T})}\right\}
\end{aligned}$$

We have found in practice that the acceptance ratio using these well-formed proposal distributions is very high (approximately 90%) when sampling the entire template and excitation matrices.

An extension of Chib's method for evaluating the marginal likelihood from the MH output [Chib and Jeliazkov, 2001] is outlined below. At any point $\{\mathbf{T}^*, \mathbf{V}^*\}$ the following holds:

$$\log p(\mathbf{X}) = \log p(\mathbf{X}, \mathbf{T}^*, \mathbf{V}^*) - \log p(\mathbf{T}^*|\mathbf{X}) - \log p(\mathbf{V}^*|\mathbf{X}, \mathbf{T}^*)$$

The posterior ordinates are given by

$$p(\mathbf{T}^*|\mathbf{X}) = \frac{\langle \alpha(\mathbf{T},\mathbf{T}^*|\mathbf{X},\mathbf{V}) q(\mathbf{T},\mathbf{T}^*|\mathbf{X},\mathbf{V}) \rangle_{p(\mathbf{T},\mathbf{V}|\mathbf{X})}}{\langle \alpha(\mathbf{T}^*,\mathbf{T}|\mathbf{X},\mathbf{V}) \rangle_{p(\mathbf{V}|X) q(\mathbf{T}^*,\mathbf{T}|\mathbf{X},\mathbf{V})}}$$

$$p(\mathbf{V}^*|\mathbf{X},\mathbf{T}^*) = \frac{\langle \alpha(\mathbf{V},\mathbf{V}^*|\mathbf{X},\mathbf{T}^*) q(\mathbf{V},\mathbf{V}^*|\mathbf{X},\mathbf{T}^*) \rangle_{p(\mathbf{V}|\mathbf{X},\mathbf{T}^*)}}{\langle \alpha(\mathbf{V}^*,\mathbf{V}|\mathbf{X},\mathbf{T}^*) \rangle_{q(\mathbf{V}^*,\mathbf{V}|\mathbf{X},\mathbf{T}^*)}}$$

for which Monte-Carlo estimates are given by

$$p(\mathbf{T}^*|\mathbf{X}) \approx \frac{M^{-1} \sum_{m=1}^{M} \alpha(\mathbf{T}^{(m)},\mathbf{T}^*|\mathbf{X},\mathbf{V}^{(m)}) q(\mathbf{T}^{(m)},\mathbf{T}^*|\mathbf{X},\mathbf{V}^{(m)})}{J^{-1} \sum_{j=1}^{J} \alpha(\mathbf{T}^*,\mathbf{T}^{(j)}|\mathbf{X},\mathbf{V}^{(j)})}$$

$$p(\mathbf{V}^*|\mathbf{X},\mathbf{T}^*) \approx \frac{J^{-1} \sum_{j=1}^{J} \alpha(\mathbf{V}^{(j)},\mathbf{V}^*|\mathbf{X},\mathbf{T}^*) q(\mathbf{V}^{(j)},\mathbf{V}^*|\mathbf{X},\mathbf{T}^*)}{G^{-1} \sum_{g=1}^{G} \alpha(\mathbf{V}^*,\mathbf{V}^{(g)}|\mathbf{X},\mathbf{T}^*)}$$

To calculate these ordinates requires three sampling runs:

1. Sample $\{\mathbf{T}^{(m)},\mathbf{V}^{(m)}\} \sim p(\mathbf{T},\mathbf{V}|\mathbf{X}), m = 1,\ldots,M$ by the MH algorithm. Select $\{\mathbf{T}^*,\mathbf{V}^*\}$ as the mode found by this sampling run for the best estimate of the marginal likelihood.

2. Sample $\{\mathbf{V}^{(j)}\} \sim p(\mathbf{V}|\mathbf{X},\mathbf{T}^*), j = 1,\ldots,J$, also generating $\{\mathbf{T}^{(j)}\} \sim q(\mathbf{T}^*,\mathbf{T}^{(j)}|\mathbf{X},\mathbf{V}^{(j)})$ after each step. This is carried out by running the MH algorithm, but rejecting all moves $\mathbf{T}^* \rightarrow \mathbf{T}^{(j)}$.

3. Sample $\mathbf{V}^{(g)} \sim q(\mathbf{V}^*,\mathbf{V}^{(g)}|\mathbf{X},\mathbf{T}^*), g = 1,\ldots,G$.

The above Metropolis-Hastings algorithm is also straightforward to apply to the Bayesian NMF model in Cemgil [2008], circumventing the multinomial sampling required for the Gibbs sampler.

7.3.2.3 Hyperparameter Optimization

Hyperparameter optimization using MCMC schemes can be carried out by a number of ways. In analogy to maximizing the variational bound as considered in Section 7.3.1.3, the Markov chain can be run for a number of iterations, and then the hyperparameters estimated from the sample statistics of the chain. However, after this step, the normalization constant $p(\mathbf{X})$ has increased, and the chain is not valid for the new hyperparameters. Either the chain has to be discarded, which is wasteful, or the samples have to be re-weighted. Re-weighting using reverse logistic regression is discussed in Geyer [1991].

An simpler method involves extending the MCMC scheme to sampling the hyperparameters based on their likelihood function, i.e., we sample from the posterior of the hyperparameters assuming flat priors, which was the case with the variational procedure. The use of flat (improper) priors does not create problems, because the calculation of the marginal likelihood is with respect to the hyperparameters, we are not integrating them out.

We use the same notation as in Section 7.3.1.3 to denote any tying structure on the hyperparameters.

The posterior distribution of the scale parameter (see (7.12)) is Gamma (Section A.2):

$$p(b_{\mathcal{M}}|\{a_m, r_m : m \in \mathcal{M}\}) \propto \exp\left(-b_{\mathcal{M}} \sum_{m\in\mathcal{M}} \frac{a_m}{r_m} + \log b_{\mathcal{M}} \sum_{m\in\mathcal{M}} a_m\right)$$
$$= \mathcal{G}\left(b_{\mathcal{M}}; 1 + \sum_{m\in\mathcal{M}} a_m, \sum_{m\in\mathcal{M}} \frac{a_m}{r_m}\right)$$

which means a Gibbs sampler step can be used to optimize the shape parameters.

The posterior distribution of the scale parameter is:

$$p(a_{\mathcal{M}}|\{b_m, r_m : m \in \mathcal{M}\}) \propto \exp\left(-a_{\mathcal{M}} \sum_{m\in\mathcal{M}} \left(\log r_m + \frac{b_m}{r_m} - \log b_m\right) + |\mathcal{M}|(a_{\mathcal{M}} \log a_{\mathcal{M}} - \log\Gamma(a_{\mathcal{M}}))\right)$$

which is not a standard distribution. Sampling from this requires a Metropolis-Hastings step.

7.3.3 Importance Sampling

Importance sampling is not suitable for practical applications of the Gaussian variance model because of the high dimensionality of the posterior. It involves computing a Monte-Carlo estimate using samples from the prior $p(\mathbf{T}, \mathbf{V})$, but without any iterations which perform some degree of source separation to update the columns of $\mathbf{T}$ and the rows of $\mathbf{V}$, almost all of the samples drawn from the prior are far from the mode and the marginal likelihood is severely underestimated.

However importance sampling may be used for single-source examples to confirm values of the marginal likelihood calculated by the other methods. The marginal likelihood can be written as the expected value of the likelihood under the prior,

$$p(\mathbf{X}) = \langle p(\mathbf{X}|\mathbf{T}, \mathbf{V})\rangle_{p(\mathbf{T},\mathbf{V})} = \int p(\mathbf{X}|\mathbf{T}, \mathbf{V}) p(\mathbf{T}, \mathbf{V}) \, \mathrm{d}\mathbf{T} \, \mathrm{d}\mathbf{V}$$

which can be approximated with the Monte-Carlo estimate

$$p(\mathbf{X}) \approx \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{X}|\mathbf{T}^{(n)}, \mathbf{V}^{(n)}) \qquad \{\mathbf{T}^{(n)}, \mathbf{V}^{(n)}\} \sim p(\mathbf{T}, \mathbf{V}), n = 1, \ldots, N$$

7.3.4 Consistency of Marginal Likelihood Estimates

We use a toy example to confirm the consistency of the marginal likelihood calculations. With $F = I = T = 1, a^t = b^t = a^v = b^v = 100$, and $\mathbf{T}, \mathbf{V}, \mathbf{X}$ set to the mode of the prior, all four methods discussed return a marginal log likelihood of -5.5266, and with $T = 2$ the marginal log likelihood is -11.0462. Both of these values are confirmed with `MATLAB` quadrature methods over the double and triple integrals respectively.

For larger models, such as those arising in musical audio analysis as described in Section 7.4, only the variational Bayes approach and the Metropolis-Hastings algorithms are practical. The VB algorithm converges to a lower bound on the marginal likelihood, because the approximating distribution to the posterior ignores the coupling between the latent sources and the templates/excitations; whilst the MH algorithm converges in the limit to the true marginal likelihood. In the single source case, the VB algorithm converges to the true

marginal likelihood (as there is no coupling between the latent sources and the templates/excitations to be ignored), but for more than one source, the VB algorithm underestimates the marginal likelihood compared with the estimate returned by the MH algorithm. However, the discrepancy between the two estimates is negligible compared with the ratios of marginal likelihood when selecting between different numbers of sources for some observed data, as described in Section 7.4.3. For Bayesian model selection, the VB and MH algorithms would lead us to the same conclusions.

7.4 Musical Audio Analysis

The Gaussian variance matrix factorization model is suitable for the time-frequency surfaces that result from applying a transformation matrix to an audio signal. The Bayesian extension is particularly useful for specifying prior knowledge concerning the spectral profile of musical instruments (templates) and volume / damping (excitations). A signal $y = (y_1, \ldots, y_n, \ldots, y_N)$ is represented by a linear combination $y_n = \sum_{\nu,\tau} \phi_n^{(\nu,\tau)} x_{\nu,\tau}$ where $\phi_n^{(\nu,\tau)}$ are localized sinusoidal basis functions in time τ and frequency ν.

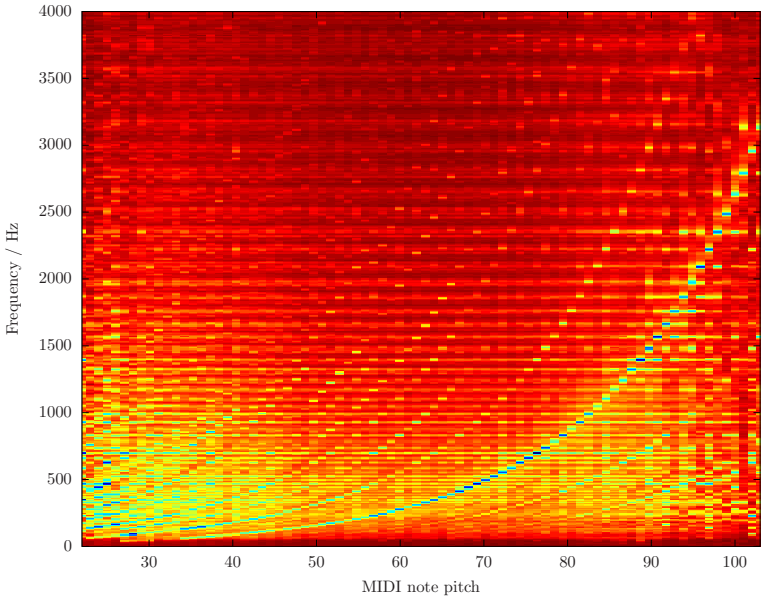
The choice of basis functions determines the transform. The short-time Fourier transform (STFT) uses time windowed complex exponentials at linearly spaced frequencies, and the resulting expansion coefficients $x_{\nu,\tau}$ are accordingly complex valued. The [short-time] discrete Cosine transform (DCT) uses even sinusoidal functions, and the expansion coefficients are real valued. Other transforms for musical audio processing include the Gabor regression model of Wolfe et al. [2004], wavelets [Mallat, 1999], the modified discrete Cosine transform (MDCT) [Daudet and Sandler, 2004] and the constant-Q transform of Brown [1991].

In the following examples, the observation matrix is formed of a matrix of DCT coefficients. Audio signals are downsampled to 8000Hz and buffered into frames of $F = 1024$ samples with no windowing or overlapping. Inference is carried out using the variational Bayes algorithm.

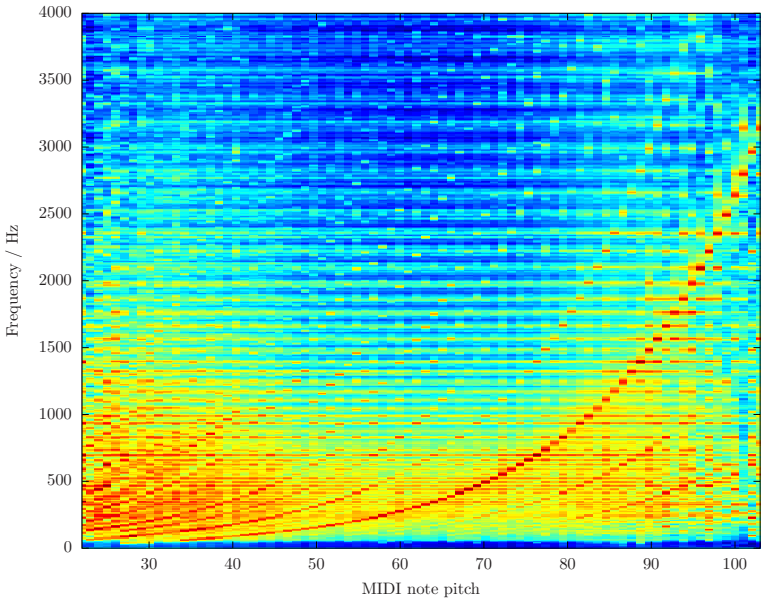
7.4.1 Model Training

Here, we optimize hyperparameters for a set of piano notes. The RWC musical instrument sounds database [Goto et al., 2003, Goto, 2004] contains audio for three pianos played with a variety of dynamics and playing styles. In Figure 7.3 on page 117 we display the resulting template hyperparameters for the audio `011PFNOF`, `011PFNOM`, and `011PFNOP` in the database, which denotes a piano played with 'normal' style at the dynamics *forte*, *mezzo* and *piano*. Each note on the keyboard is played once, covering a range of 88 notes from MIDI 21 to MIDI 108. Here, individual shape and scale parameters are trained for each frequency bin for each pitch class. The plot of the scale parameters in Figure 7.3b on page 117 clearly shows the harmonic series of each note, and that the spacing between the harmonics increases with pitch. A careful inspection of the plot of the shape parameters in Figure 7.3a on page 117 shows that frequency bins corresponding to the harmonic series have a larger variance than those corresponding to the noise floor.

For this example, we have chosen to train the hyperparameters for single source models, so that it can be clearly illustrated that the priors capture the harmonic series of the piano notes. As the samples are of differing length, we choose to tie the excitation parameters across time, thus estimating a single value of a^v and b^v per note. This means that the priors estimated here are valid for signals of arbitrary length.



(a) Shape parameters $a^t_{\nu,i}$



(b) Scale parameters $b^t_{\nu,i}$

Figure 7.3: Template hyperparameters for single source models of piano notes

7.4.2 Source Separation and Visualization

Here we illustrate a source separation application using the Gaussian variance matrix factorization model. We have taken an extract of a piano piece and synthesized the MIDI file using the same audio samples in 7.4.1. Over the extract we have assumed that the model is stationary, and all 88 sources, corresponding to every note on the piano keyboard, are active. We then infer $\mathbf{V}$ using the variational Bayes algorithm. Our intuition is that notes which are being played will have a large excitation, while notes which are not being played will have a small excitation and thus be indistinguishable from silence. This is confirmed in Figure 7.4 on page 119, which is a useful alternative to frequency representations such as the harmonic transform Zhang et al. [2004], and can be used to visualize the frequency content of audio signals. All 88 possible notes are modelled using rank one source matrices, and the hyperparameters optimized separately. Regions of high excitation (in red) correspond to a note being played. The positions of the notes are offset slightly so that the high excitation regions are not obscured.

7.4.3 Model Selection

In the previous two sections, we have modelled each piano note using a single source model. It remains to be discussed however, if we would obtain a better model by using a multiple source model, i.e., a rank $I > 1$ source matrix. The goal of this section is to determine whether the variational Bayes algorithm gives consistent answers as to the optimal number of source for the piano notes.

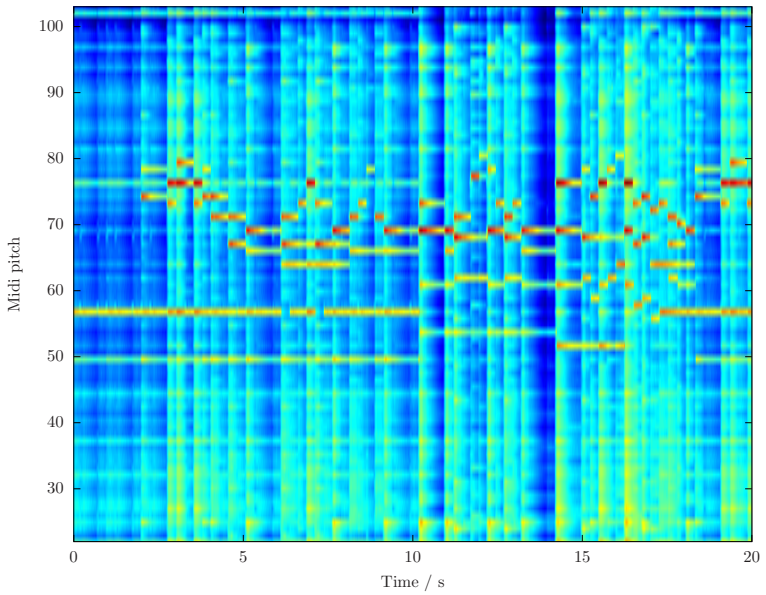
The results of our investigations are given in Figure 7.5 on page 120 for the set of piano notes `01[1,3] PFNO [F,M,P]` in the RWC database. The optimal number of sources is correlated with the pitch of the piano note, which may be the result of the particular time-frequency representation chosen for the audio. The trend shown is that notes with a small or high pitch are best modelled by a few sources, whilst notes with a medium pitch are best modelled by a larger number of sources. Factors that contribute to this are perhaps: 1) poor resolution of the DCT for low pitches 2) downsampling to 8000Hz cuts off many of the harmonics of higher pitches, thus leading to simpler models. The dependency of the number of sources on the length of the frame was not however investigated here, and is suggested for investigation in future work when multiple source models are used for polyphonic music transcription.

7.5 Prior Model for Polyphonic Piano Music

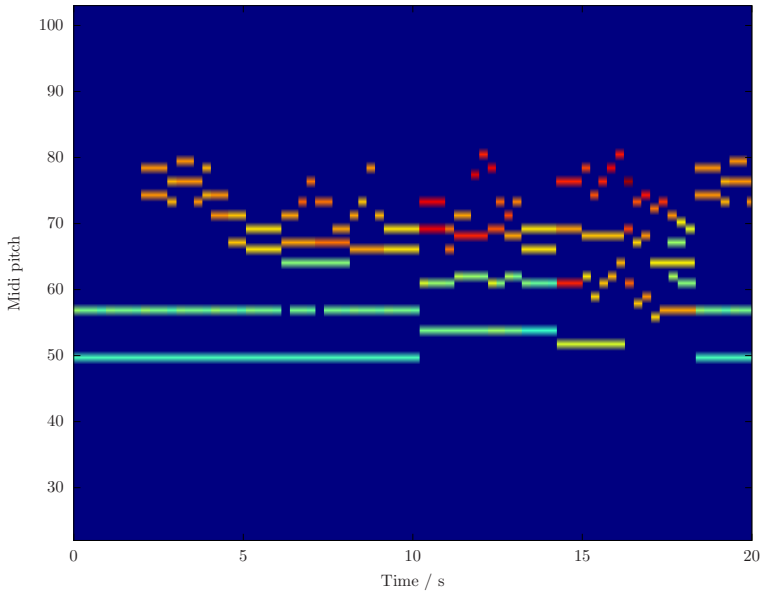
In this section, we extend the prior model for the excitation matrix to include MIDI pitch and velocity of the notes that are playing in a piece of solo polyphonic piano music. We also apply this prior model to the Poisson intensity model of Cemgil [2008] so that the transcription performance of both models can be compared.

7.5.1 Model Description

In this section, we have chosen to rely on deterministic approaches to solve the transcription inference problem, as opposed to more expensive MCMC approaches. We describe a quite general approach which lends itself to any form of music for which the MIDI format is an admissible representation of the transcription.

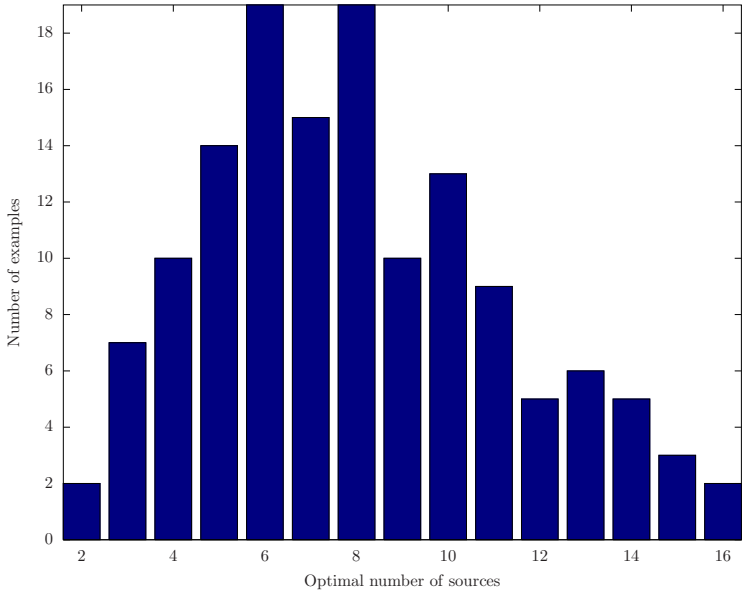


(a) Visualization by source separation. Red areas denote regions of higher excitation, which occur particularly at note onsets for notes within the melodic line. The onset excitation corresponds with the MIDI note velocity below.

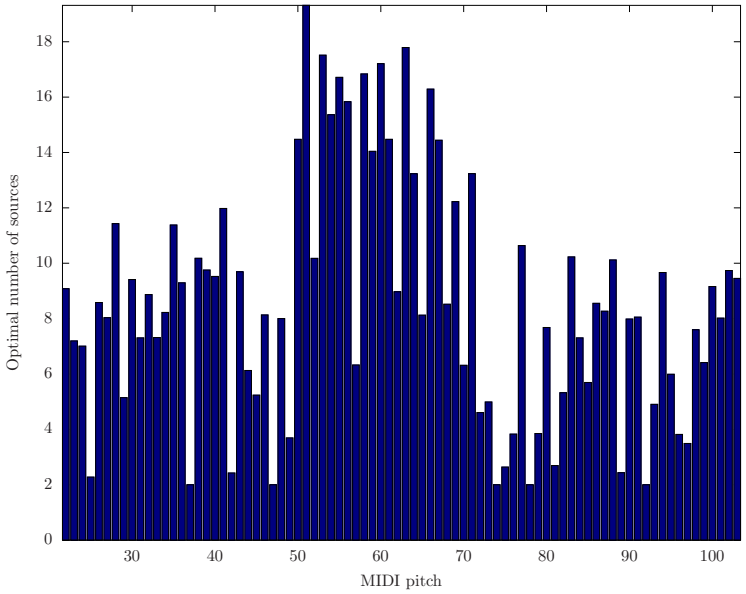


(b) Original MIDI. High note onset velocities are shaded red.

Figure 7.4: Transcription of the first 20 seconds of Albeniz's Suite Española No.5 *Asturias (Leyenda)* using the Gaussian variance matrix factorization model.



(a) Histogram of the optimal number of sources



(b) Optimal number of sources by pitch

Figure 7.5: Optimal number of sources for a set of piano notes

We select the maximum number of sources N to be the total number of pitches represented in the MIDI format. Each source i corresponds to a particular pitch. Then we have a single set of template parameters $\mathbf{T} \in \mathbb{R}_+^{F \times I}$ for all I sources, which are intended to represent the spectral, harmonic information of the pitches.

For polyphonic transcription, we are typically interested in inferring the piano roll matrix $\mathbf{C}$ which owing to the above assumption of one source per pitch has the same dimensions as the excitation matrix $\mathbf{V}$. For note i at time τ we set $\mathbf{C}_{n,\tau}$ to be the value of the velocity of the note, and $\mathbf{C}_{n,\tau} = 0$ if the note is not playing. We use the NOTE ON velocity, which is stored in the MIDI format as a integer between 1 and 128. Thus, we model note velocity using our generative model. This contrasts with previous approaches which infer a binary-valued piano roll matrix of note activity, essentially discarding potentially useful volume information. The prior distribution $p(\mathbf{C})$ is a discrete distribution, which can incorporate note transition probabilities and commonly occurring groups of note pitches, i.e., chords and harmony information.

A note with a larger velocity will have a larger corresponding excitation. The magnitude of the excitation will depend on the pitch of the note as well as its velocity, so instead of applying a volume curve such as (2.1), we infer this relationship from training data. We will represent this information as an *a priori* unknown positive-valued matrix $\mathbf{F}$ of size $I \times 128$ where $\mathbf{F}_{i,z}$ represent a mapping from the MIDI pitch i and velocity z to the excitation matrix given by

$$\mathbf{V}_{i,\tau} = \begin{cases} 0 & \mathbf{C}_{i,\tau} = 0 \\ \mathbf{F}_{i,\mathbf{C}_{i,\tau}} & \text{otherwise} \end{cases} \tag{7.15}$$

For music transcription, we extend the prior model on $\mathbf{V}$ to include the matrices $\mathbf{F}$and $\mathbf{C}$, i.e.,

$$p(\mathbf{V}, \mathbf{F}, \mathbf{C}) = p(\mathbf{V}|\mathbf{F}, \mathbf{C})p(\mathbf{F}, \mathbf{C})$$

As $\mathbf{F}$ is a mapping to the excitation matrix, we place an inverse-gamma prior (for the Gaussian variance model) or a gamma prior (for the Poisson intensity model) over each element of $\mathbf{F}$. The resulting conditional posterior over $\mathbf{F}$ is of the same family as the prior, and is obtained by combining the expectations of the sources corresponding to the correct pitch and velocity.

The full generative model for polyphonic transcription is given by

$$p(\mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{V}, \mathbf{F}, \mathbf{C}) = p(\mathbf{X}|\mathbf{S})p(\mathbf{S}|\mathbf{T}, \mathbf{V})p(\mathbf{V}|\mathbf{F}, \mathbf{C})p(\mathbf{T})p(\mathbf{F}, \mathbf{C})$$

One advantage of this model is that that minimal storage is required for the parameters which can be estimated offline from training data, as we demonstrate in 7.6.2. The two sets of parameters are intuitive for musical signals. This model also allows closer modeling of the excitation of the notes that the MIDI format allows.

7.5.2 Algorithm

We are able to integrate out the latent sources $\mathbf{S}$ (see Section 7.2), and also eliminate $\mathbf{V}$ given $\mathbf{F}$ and $\mathbf{C}$, using (7.15). The algorithm we present here is a generalized EM algorithm, which iterates to find a solution of the posterior:

$$\arg \max_{\mathbf{T},\mathbf{F},\mathbf{C}} p(\mathbf{T}, \mathbf{F}, \mathbf{C}|\mathbf{X})$$

The posterior distribution of $\mathbf{F}$ conditional on $\mathbf{C}, \mathbf{T}, \mathbf{X}$ is inverse-gamma as it is formed by collecting the estimates of $\mathbf{V}$ corresponding to each note pitch/velocity pairing.

To maximize for the piano roll $\mathbf{C}$ we first note that each frame of observation data is independent given the other parameters $\mathbf{V}, \mathbf{F}$. For each τ we wish to calculate

$$\arg\max_{\mathbf{C}_\tau} p(\mathbf{X}_\tau|\mathbf{T}, \mathbf{V}_\tau)p(\mathbf{V}_\tau|\mathbf{F}, \mathbf{C}_\tau)p(\mathbf{F}, \mathbf{C}_\tau) \tag{7.16}$$

where $\mathbf{X}_\tau, \mathbf{V}_\tau$ and $\mathbf{C}_\tau$ are the τth column vectors of $\mathbf{X}, \mathbf{V}$ and $\mathbf{C}$ respectively. However, as each $\mathbf{C}_\tau$ has 128^I possible values, an exhaustive search to maximize this is not feasible. Instead, we have found that the following greedy search algorithm works sufficiently well: for each frame τ calculate

$$\arg\max_{\tilde{\mathbf{C}}_\tau} p(\mathbf{X}_\tau|\mathbf{T}, \tilde{\mathbf{V}}_\tau)p(\tilde{\mathbf{V}}_\tau|\mathbf{F}, \tilde{\mathbf{C}}_\tau)p(\mathbf{F}, \tilde{\mathbf{C}}_\tau) \tag{7.17}$$

where $\tilde{\mathbf{C}}_\tau$ differs from $\mathbf{C}_\tau$ by at most one element, and $\tilde{\mathbf{V}}$ is the corresponding excitation matrix. There are $I \times 128$ possible settings of $\tilde{\mathbf{C}}_\tau$ for which we evaluate the likelihood at each stage of the greedy search. This can be carried out efficiently by noticing that during the search the corresponding matrix products $\mathbf{T}\tilde{\mathbf{V}}$ differ from the existing value by only a rank-one update of $\mathbf{TV}$.

The resulting algorithm has one update for the expectation step and three possible updates for the maximization step. For the generalized EM algorithm to be valid, we must ensure that any maximization step based on parameter values not used to calculate the source expectations is not guaranteed to increase the log likelihood, and therefore must be verified.

7.6 Results

A useful comparative study of three differing approaches has been carried out in Poliner and Ellis [2007]. A dataset with ground-truth of polyphonic piano music has been provided to assess the performance of a support-vector machine (SVM) classifier, [Poliner and Ellis, 2007], which is provided as an example of a discriminative based approach, having favorable performance in classification accuracy; a neural-network classifier Marolt [2004], known as SONIC[2]; and an auditory-model based approach Ryynänen and Klapuri [2005].

7.6.1 Comparison

To comprehensively evaluate these models, we use Poliner and Ellis training and test data and compare the performance against the results provided in the same paper, which are repeated here for convenience. The ground truth for the data consists of 124 MIDI files of classical piano music, of which 24 have been designated for testing purposes and 13 are designated for validation. In a Bayesian framework there need not be any distinction between training and validation data: both may be considered labeled observations. Here we have chosen to discard the validation data rather than include it in the training examples for a fairer comparison with the approaches used by other authors. We also do not attempt to optimize the model

[2] `http://lgm.fri.uni-lj.si/SONIC`

Algorithm 7.2 Gaussian Variance: algorithm for polyphonic transcription

- **Source Expectation**

$$\left\langle \mathbf{s}_{\nu,\tau}\mathbf{s}_{\nu,\tau}^{\top}\right\rangle = [\mathbf{t}_{\nu}\mathbf{v}_{\tau}^{\top}]\cdot I_N - \kappa_{\nu,\tau}\kappa_{\nu,\tau}^{\top}[\mathbf{TV}]_{\nu,\tau} + \left\langle \mathbf{s}_{\nu,\tau}\right\rangle \left\langle \mathbf{s}_{\nu,\tau}\right\rangle^{\top}$$

- **Template Maximization**
  Shape and scale parameters of inverse-gamma posterior distribution

$$\begin{aligned} A_{\nu,i} &= \alpha_{\nu,i}^{(\mathbf{T})} + T \\ B_{\nu,i} &= \beta_{\nu,i}^{(\mathbf{T})} + \sum_{\tau}\mathbf{V}_{n,\tau}^{-1}\left\langle \mathbf{s}_{\nu,k}\mathbf{s}_{\nu,\tau}^{\top}\right\rangle \end{aligned}$$

  Mode of posterior distribution

$$\mathbf{T}_{\nu,i} \leftarrow \frac{B_{\nu,i}}{A_{\nu,i}+1}$$

- **Excitation Maximization**
  Shape and scale parameters of inverse-gamma posterior distribution

$$\begin{aligned} A_{i,z} &= \sum_{\{\tau:\mathbf{C}_{i,\tau}=z\}} \alpha_{n,\tau}^{(\mathbf{V})} + \mathbf{F}_{i,\mathbf{C}_{i,\tau}} \\ B_{i,z} &= \sum_{\{\tau:\mathbf{C}_{i,\tau}=z\}} \Big(\beta_{n,\tau}^{(\mathbf{V})} + \sum_{\nu}\mathbf{T}_{\nu,i}^{-1}\left\langle \mathbf{s}_{\nu,\tau}\mathbf{s}_{\nu,\tau}^{\top}\right\rangle\Big) \end{aligned}$$

  Mode of posterior distribution

$$\mathbf{F}_{i,z} \leftarrow \frac{B_{i,z}}{A_{i,z}+1}$$

- **Transcription Search**
  for $\tau = 1,\ldots,T$

$$\mathbf{C}_{\tau} \leftarrow \arg\max_{\tilde{\mathbf{C}}_{\tau}} \sum_{\nu}\Big(-\frac{1}{2}\frac{|\mathbf{X}_{\nu,\tau}|^2}{[\mathbf{T}\tilde{\mathbf{V}}]_{\nu,\tau}} - \log[\mathbf{T}\tilde{\mathbf{V}}]_{\nu,\tau}\Big)p(\mathbf{F},\tilde{\mathbf{C}}_{\tau})$$

Algorithm 7.3 Poisson Intensity: algorithm for polyphonic transcription

- **Source Expectation**

$$\langle \mathbf{s}_{\nu,\tau} \rangle = \kappa_{\nu,\tau} \mathbf{X}_{\nu,\tau}$$

- **Template Maximization**
  Shape and scale parameters of inverse-gamma posterior distribution

$$\begin{aligned} A_{\nu,i} &= \alpha^{(\mathbf{T})}_{\nu,i} + \sum_{\tau} \langle \mathbf{s}_{i,\tau} \rangle \\ B_{\nu,i} &= \beta^{(\mathbf{T})}_{\nu,i} + \sum_{\tau} \mathbf{V}_{i,\tau} \end{aligned}$$

  Mode of posterior distribution

$$\mathbf{T}_{\nu,i} \leftarrow \frac{A_{\nu,i} - 1}{B_{\nu,i}}$$

- **Excitation Maximization**
  Shape and scale parameters of inverse-gamma posterior distribution

$$\begin{aligned} A_{i,z} &= \sum_{\{\tau : \mathbf{C}_{i,\tau} = z\}} \Big(\alpha^{(\mathbf{V})}_{i,\tau} + \sum_{\nu} \langle \mathbf{s}_{\nu,\tau} \rangle \Big) \\ B_{i,z} &= \sum_{\{\tau : \mathbf{C}_{i,\tau} = z\}} \Big(\beta^{(\mathbf{V})}_{i,\tau} + \sum_{\nu} \mathbf{T}_{\nu,i} \Big) \end{aligned}$$

  Mode of posterior distribution

$$\mathbf{F}_{i,z} \leftarrow \frac{A_{i,z} - 1}{B_{i,z}}$$

- **Transcription Search**
  for $\tau = 1, \ldots, T$

$$\mathbf{C}_{\tau} \leftarrow \arg\max_{\tilde{\mathbf{C}}_{\tau}} \sum_{\nu} \Big(\mathbf{X}_{\nu,\tau} \log[\mathbf{T}\tilde{\mathbf{V}}]_{\nu,\tau} - [\mathbf{T}\tilde{\mathbf{V}}]_{\nu,\tau} \Big) p(\mathbf{F}, \tilde{\mathbf{C}}_{\tau})$$

parameters to minimize transcription errors on the validation set, as this is not consistent with a generative modelling approach. This is discussed further in Section 7.7.

The observation data is primarily obtained by using a software synthesizer to generate audio data. In addition, 19 of the training tracks and 10 of the test tracks were synthesized and recorded on a Yamaha Disklavier. Only the first 60 seconds of each extract is used. The audio, sampled at 8000 Hz, is then buffered into frames of length 128 ms with a 10ms hop between frames, and the spectrogram is obtained from the short-time Fourier transform of these frames. Poliner and Ellis subsequently carry out a spectral normalization step in order to remove some of the timbral and dynamical variation in the data prior to classification. However, we omit this processing stage as we rather wish to capture this information in our generative model.

7.6.2 Implementation

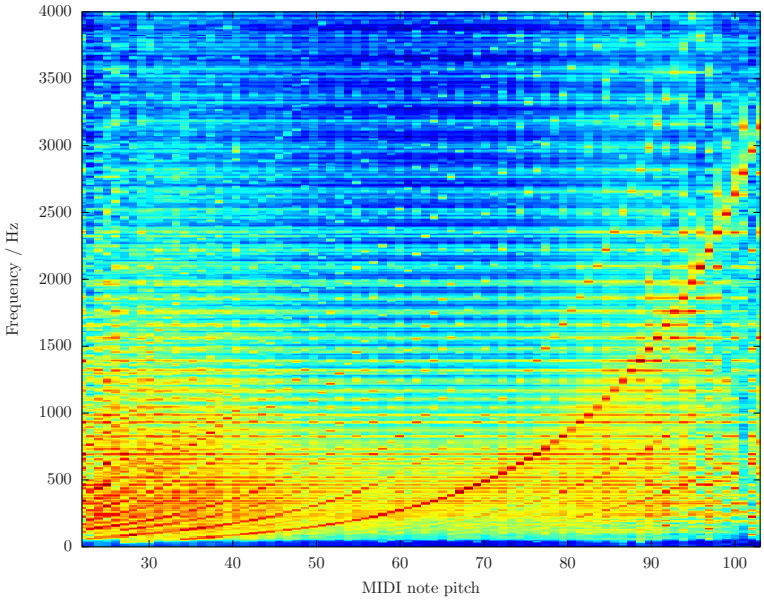
Because of the copious amount of training data available, there is enough information concerning the frequencies of the occurrence of the note pitches and velocities that it is not necessary to place informative priors on these parameters.

It is not necessary to explicitly carry out a training run to estimate values of the model parameters before evaluating against the test data. However the EM algorithm does converge faster during testing if we first estimate the parameters from the labelled observations. Figure 7.6 on page 126 and Figure 7.7 on page 127 show the $\log \mathbf{T}$ templates and $\log \mathbf{F}$ excitation parameters estimated from the Poliner and Ellis training data for the Gaussian variance and Poisson intensity models, with flat prior distributions after running the EM algorithm to convergence on the training data only, and using a single source to model each note pitch as in 7.4.1. The templates clearly exhibit the harmonic series of the musical notes, and the excitations contain the desired property that notes with higher velocity correspond to higher excitation, hence our assumption of uniform priors on these parameters has not been detrimental. For the excitation parameters, white areas denote pitch/velocity pairs that are not present in the training data and are thus unobserved.

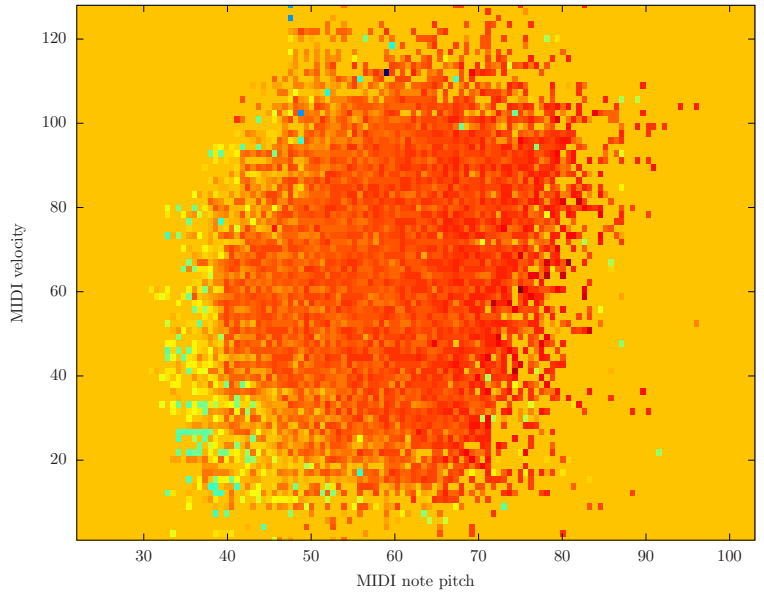
For each of the matrix factorization models we consider two choices of the prior $\mathbf{C}$. The first assumes that each frame of data is independent of the others, which is useful in evaluating the performance of the source models in isolation. The second assumes that each note pitch is independent of the others, and between consecutive frames there is a state transition probability, where the states are each note being active or inactive, i.e.,

$$p(\mathbf{C}_{i,\tau} > 0 | \mathbf{C}_{i,\tau-1} = 0) = p(\mathbf{C}_{i,\tau} = 0 | \mathbf{C}_{i,\tau-1} > 0) = p_{\text{event}} \tag{7.18}$$

This prior is known as the Markov prior in the remainder of this chapter. The state transition probabilities are estimated from the training data. It is possible and more correct to include these transition probabilities as parameters in the model, but we have not carried out the inference of note transition probabilities in this work. Similar Markov time dependencies between frames of data modelled by NMF techniques are used in Ozerov et al. [2009]. The modification to the inference is straightforward: in (7.16) the prior on $\mathbf{C}_{i,\tau}$ is calculated using (7.18) using the current values of $\mathbf{C}_{i,\tau-1}$ and $\mathbf{C}_{i,\tau+1}$ that have been estimated.

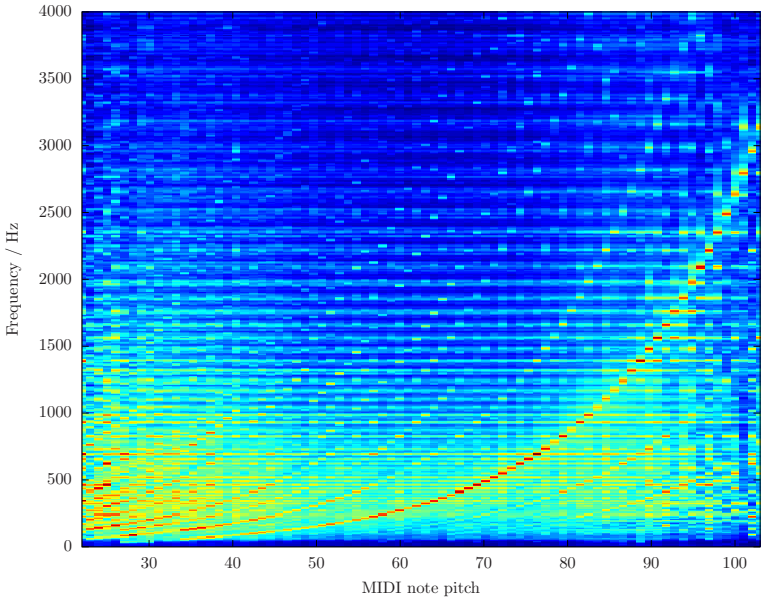


(a) Template parameters $\log \mathbf{T}$

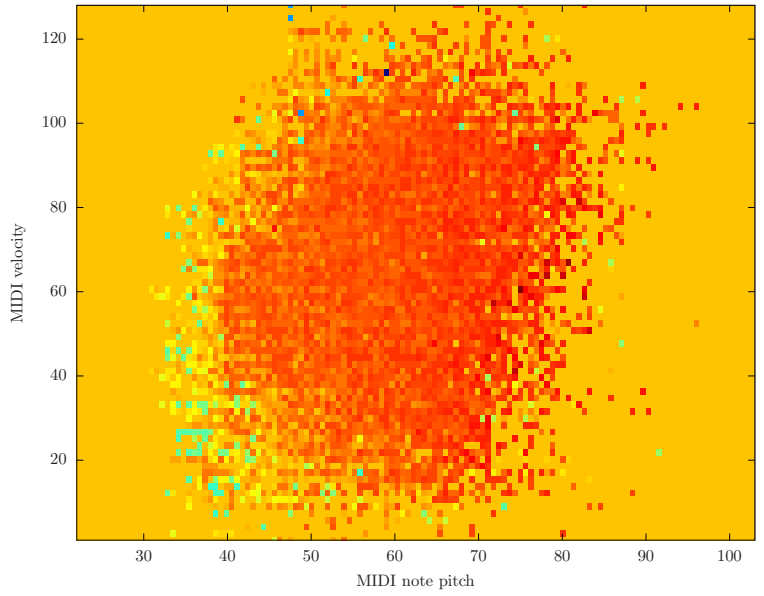


(b) Velocity-Excitation Mapping $\log \mathbf{F}$

Figure 7.6: Parameter estimates for the Gaussian variance model from training data



(a) Template parameters $\log \mathbf{T}$



(b) Velocity-Excitation Mapping $\log \mathbf{F}$

Figure 7.7: Parameter estimates for the Poisson model from training data

7.6.3 Evaluation

Following training, the matrix of spectrogram coefficients is then extended to include the test extracts. As the same two instruments are used in the training and test data, we simply use the same parameters which were estimated in the training phase. We transcribe each test extract independently of the others, yet note that in the full Bayesian setting this should be carried out jointly, however this is not practical or typical of a reasonable application of a transcription system. An example of the transcription output for the first ten seconds of the synthesized version of Burgmueller's *The Fountain* is provided for the Gaussian variance model, both with independent and Markov priors (Figure 7.8 on page 129 and Figure 7.9 on page 130) on $\mathbf{C}$, compared to the MIDI ground truth (Figure 7.10 on page 131). The transcription is graphically represented in terms of detections and misses in Figure 7.11 on page 132. True positives are in light gray, false positives in dark gray, and false negatives in black. Most of the difficulties encountered in transcription in this particular extract were due to the positioning of note offsets, rather than the detection of the pitches themselves.

The transcription with independent prior on $\mathbf{C}$ shows that the generative model has not only detected the activity of many of the notes playing, but also has attempted to jointly infer the velocity of the notes. Each frame has independently inferred velocity, hence there is much variation across a note, however there is correlation between the maximum inferred velocity during a note event and the ground truth velocities. The Markov prior on $\mathbf{C}$ eliminates many of the spurious notes detected, which are typically of a short duration of a few frames.

We have used only the information contained in note pitches, but the effect of resonance and pedaling can be clearly seen by comparing the ground truth with the transcriptions. This motivates the use of a note onset evaluation criteria.

We follow the same evaluation criteria as provided by Poliner and Ellis. As well as recording the accuracy Acc (true positive rate), the transcription is error is decomposed into three parts: Subs the substitution error rate, when a note from the ground truth is transcribed with the wrong pitch; Miss the note miss rate, when a note in the ground truth is not transcribed, and FA the false alarm rate beyond substitutions, when a note not present in the ground truth is transcribed. These sum to form the total transcription error Tot which cannot be biased simply by adjusting a threshold for how many notes are transcribed.

Table 7.1 on page 129 shows the frame-level transcription accuracy for the approaches studied in Poliner and Ellis [2007]. We are using the same data sets and features dimensions selected by the authors of this paper to compare our generative models against these techniques. This table expands the accuracy column in Table 7.2 on page 129 by splitting the test data into the recorded piano extracts and the MIDI synthesized extracts.

Table 7.2 on page 129 shows the frame-level transcription results for the full synthesized and recorded data set. Accuracy is the true positive rate expressed as a percentage, which can be biased by not reporting notes. The total error is a more meaningful measure which is divided between substitution, note misses and false alarm errors. This table shows that the matrix factorization models with a Markov note event prior have a similar error rate to the Marolt system on this dataset, but has a greater error rate than the support vector machine classifier.

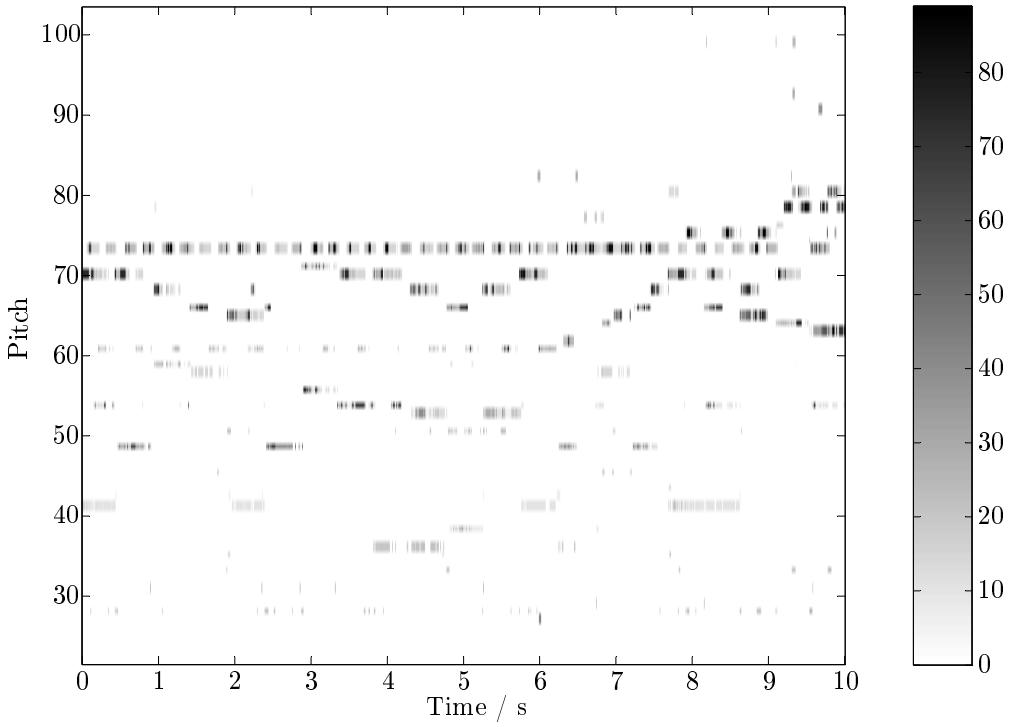


Figure 7.8: Transcription using *a priori* independent frames

Table 7.1: Frame-level transcription accuracy

Model	Piano	MIDI	Both
SVM	56.5	72.1	67.7
Ryynänen & Klapuri	41.2	48.3	46.3
Marolt	38.4	40.0	39.6
Variance (Independent)	36.0	41.2	39.7
Variance (Markov)	38.0	44.0	42.3
Intensity (Independent)	40.1	35.4	36.8
Intensity (Markov)	39.7	36.2	37.3

Table 7.2: Frame-level transcription results

Model	Acc	Tot	Subs	Miss	FA
SVM	67.7	34.2	5.3	12.1	16.8
Ryynänen & Klapuri	46.6	52.3	15.0	26.2	11.1
Marolt	36.9	65.7	19.3	30.9	15.4
Variance (Independent)	39.7	68.2	22.9	27.7	17.6
Variance (Markov)	42.3	62.1	18.1	32.0	12.0
Intensity (Independent)	36.8	71.0	27.8	24.6	18.6
Intensity (Markov)	37.3	66.6	23.7	30.0	12.9

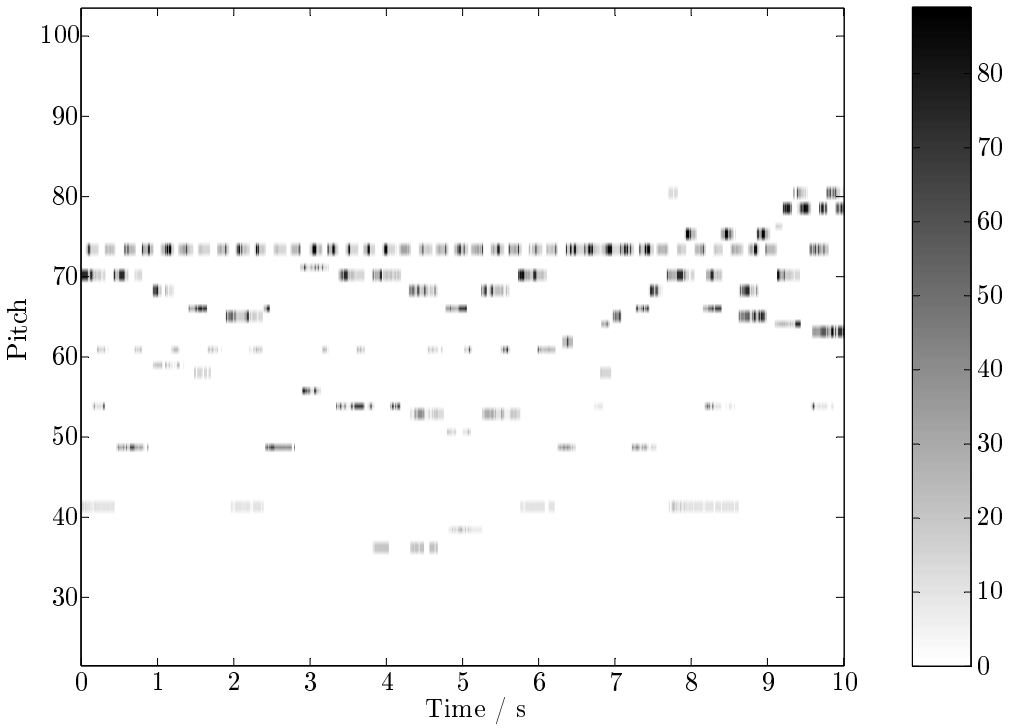


Figure 7.9: Transcription using Markov transition probabilities between frames

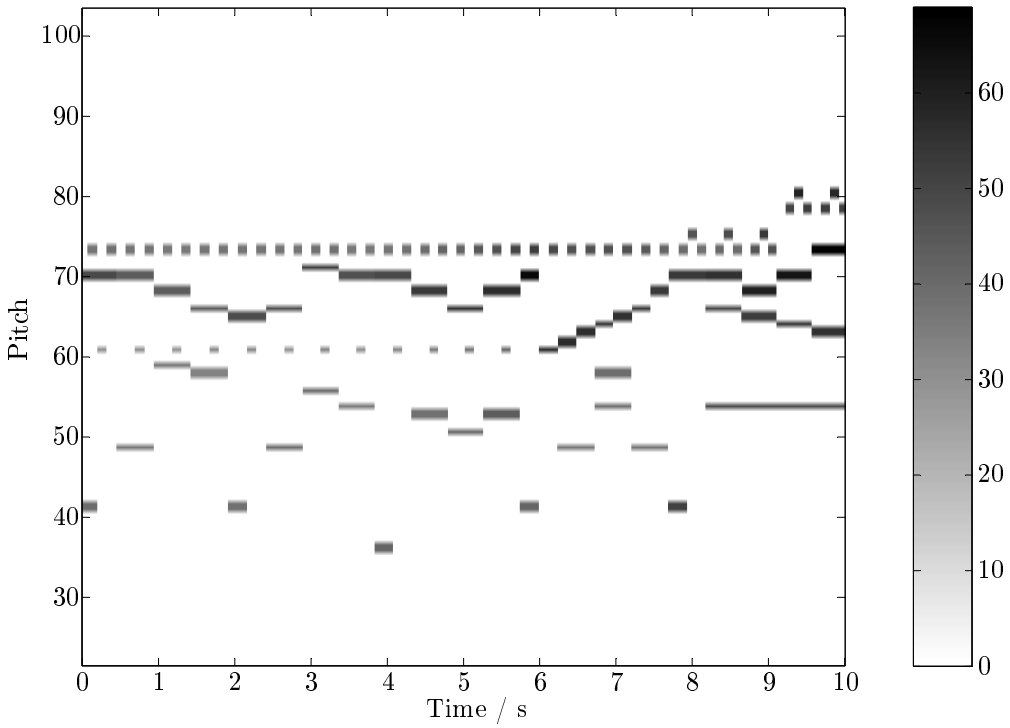


Figure 7.10: Ground truth for the transcription results in Figure 7.8 on page 129 and Figure 7.9 on page 130

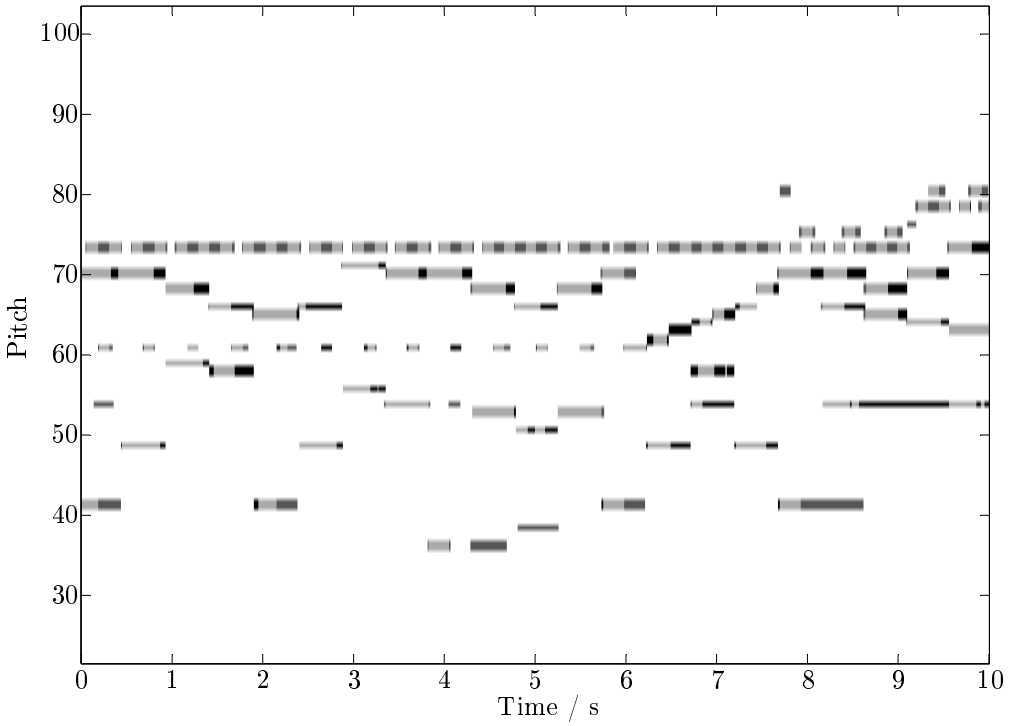


Figure 7.11: Detection assessment

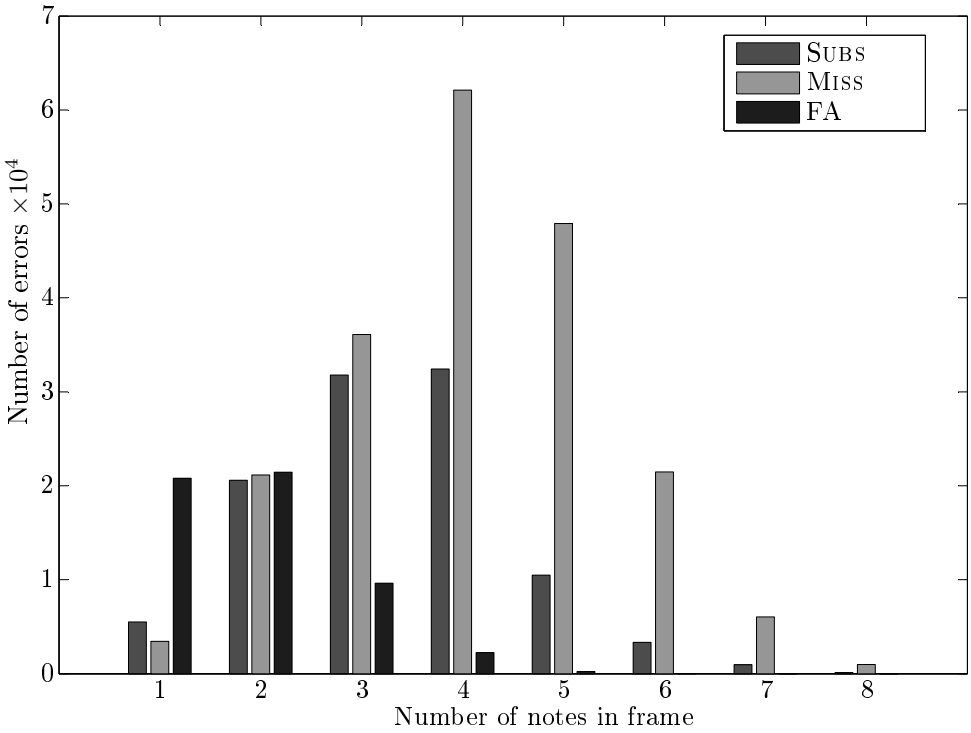


Figure 7.12: Number of errors for the Gaussian variance Markov model by number of notes and error type.

7.7 Conclusion

In this chapter we have described a generative model for factorizing the variances of matrix elements into smaller template and excitation matrices, and developed Bayesian priors and inference algorithms to estimate the variances and the dimensions of the factor matrices. When applied to the spectrogram of a musical note, the template matrix models the harmonic content of the note, and the excitation matrix controls how the volume of the note varies over time. The hyperparameter optimization techniques described in this chapter can be applied to labeled training data to develop a system capable of distinguishing between and modelling different musical instruments and pitches. We have demonstrated how the excitation matrix of a polyphonic signal can be used to visualize the transcription of the music.

We have compared the performance of generative spectrogram factorization models with three existing transcription systems on a common dataset. The models exhibit a similar error rate as the neural-network classification system of Marolt [2004]. However the support vector machine classifier of Poliner and Ellis [2007] achieves a lower error rate for polyphonic piano transcription on this dataset. In this conclusion, we principally discuss the reasons for the difference in error rate of these systems, and how the generative models can be improved in terms of inference and prior structure to achieve an improved performance.

The support vector machine is purely a classification system for transcription, for which the parameters have been explicitly chosen to provide the best transcription performance on a validation set; whereas the spectrogram factorization models, being generative in nature, are applicable to a much wider range of problems: source separation, restoration, score-audio alignment and so on. For this reason, we have not attempted to select priors by hand-tuning in order to improve transcription performance, but rather adopt a fully Bayesian approach with an explicit model which infers correlations in the spectrogram coefficients in training and test data, and thus as a product of this inference provides a transcription of the test data. The differences in this style of approach, and the subsequent difference in performance, resemble that of supervised and unsupervised learning in classification. Thus in light of this, we consider the performance of the spectrogram factorization models to be encouraging, as they are comparable to an existing polyphonic piano transcription system without explicitly attempting to improve the transcription performance by tuning prior hyperparameters. Vincent et al. [2008] for instance demonstrate the improvement in performance for polyphonic piano transcription that can be achieved over the standard NMF algorithm by developing improved basis spectra for the pitches, and achieve a performance mildly better than the neural-network classifier: a similar result to what has been presented here.

Bertin et al. [2009b] similarly report improvement in transcription performance for the Gaussian variance model compared to existing Bayesian NMF. They also suggest an tempering approach to avoid iterative algorithms being trapped in local maxima of the likelihood function. There are a number of alternative algorithms to perform NMF with the aim of increasing speed of convergence and locating better solutions. Recent work includes a split-gradient method developed by Lantéri et al. [2010].

To improve performance for transcription in a Bayesian spectrogram factorization, we can firstly improve initialization using existing multiple frequency detection systems for spectrogram data, and extend the hierarchical model for polyphonic transcription using concepts such as chords and keys. We can also jointly track tempo and rhythm using a probabilistic model, for examples of this see Whiteley et al. [2006], Raphael [2004], Peeling et al. [2007a] where the model used could easily be incorporated into the Bayesian hierarchical approach here.

The models we have used have assumed that the templates and excitations are drawn independently from priors, however the existing framework of gamma Markov fields developed in Cemgil et al. [2007], Dikmen and Cemgil [2010] can be used as replacements of these priors, and allows us to model stronger correlations, for example, between the harmonic frequencies of the same musical pitch, which additionally contain timbral content, and also model the damping of the excitation of notes from one frame to the next. It has qualitatively shown that using gamma Markov field priors results in a much improved transcription, and in future work we will use this existing framework to extend the model described in this paper, expecting to see a much improved transcription performance by virtue of a more appropriate model of the time-frequency surface. An alternative framework for enforcing temporal continuity in Bayesian NMF for polyphonic music transcription is presented by Bertin et al. [2009a], which could also be applied to the Gaussian variance model as opposed to the Poisson intensity model used by the authors. Other priors enforcing correlations between the elements in the factored matrices are possible, for example Gaussian process priors [Schmidt and Laurberg, 2008].

On this dataset, the Gaussian variance model has better performance for transcription than the intensity based model, and we suggest that this is due to the generative model modeling the weighting of the spectrogram coefficients directly, and thus being a more appropriate model for time-frequency surface estimation. However, most of the literature for polyphonic music transcription systems using matrix factorization models has focused on the KL divergence and modifications and enhancements of the basic concept. Therefore it would be useful to firstly evaluate such variants of NMF against this dataset and other systems used for comparing and evaluating music transcription systems. Secondly, it would also be useful to replace the implicit Poisson intensity source model in these approaches with the Gaussian variance model.

In summary, we have presented matrix factorization models for spectrogram coefficients using a Gaussian variance parametrization, and have developed inference algorithms for the parameters of these models. The suitability of these models has been assessed for the polyphonic transcription of solo piano music, resulting in a performance which is comparable to some existing transcription systems. As we have used a Bayesian approach, we can extend the prior structure in a hierarchical manner to improve performance and model higher-level features of music.

Chapter 8

A Probabilistic Framework for Inferring Temporal Structure in Music

In this chapter we develop a probabilistic framework for the tractable inference of temporal structure in musical audio. The goal of this framework is to unify otherwise separate applications of Bayesian musical signal processing into a common, generative modelling framework for inference. We model the performance of a piece of music as the movement of a *score pointer* through a symbolic representation of the music. This representation may be the actual written score of the piece of music being played, converted to a suitable format, when the application is score following or tracking; or the representation may be a code book of rhythmic patterns for a tempo and beat tracking application; or a code book of chords for a transcription application. The observed audio itself is modelled by a generative process conditional on the properties of the score at that point.

In the previous chapters of this thesis, we have mainly focused on processing individual frames of musical audio to detect multiple pitches with prior information. In Chapter 7 we added a simple Markov model for note transitions, such that the probability of a note sounding or ceasing to sound in a particular frame is dependent on the previous frame. This addition has a smoothing effect on the transcription such that spurious note detections are avoided, and notes are transcribed to their full length. In this chapter we define a Markov model over all of the notes sounding in each frame, mapping structures in the score, or the expected score of the music into Bayesian priors. This allows a richer, more realistic transcription of the musical structure than a simple frame-by-frame transcription would produce, and provides accuracy and robustness when aligning a preexisting transcription to a performed piece of music.

8.1 Audio Matching using Generative Models

8.1.1 Existing Dynamic Time Warping Approach

We introduce our model by considering an existing state-of-the-art approach [Hu et al., 2003, Orio and Schwarz, 2001, Turetsky and Ellis, 2003] to the alignment of two pieces of music which are assumed to share a common score. Each piece of music is buffered into overlapping frames, and a feature vector is extracted

from each frame. A distance metric is used to compute the similarity between pairs of feature vectors from each piece, and dynamic time warping (DTW) is used to compute a joint path through both pieces, maximizing the similarity between matched frames. Many choices of feature vector and distance metrics have been proposed, including chroma vectors using non-negative matrix factorization divergence measures by Niedermayer [2009]. An interesting variation of these methods is that of Stark and Plumbley [2010] which performs localized self-alignment to detect repetitious structures in music to 'follow' a performance without a score.

This approach to audio alignment may be extended to score matching and transcription by using a synthesizer to generate audio from a score or code book of musical chords. The synthesized audio is aligned to the observed audio and the path through the observed audio can be used to infer the score position or transcription. For transcription, the path must allow movements from the end of one chord to the beginning of the next in the code book. Such jumps in the path reduce the efficiency of most DTW algorithm implementations.

In this section we state the audio alignment application using a hidden Markov model for the path and a generative signal model for each frame. This allows a number of extensions to the DTW approach, for example being able to jointly align multiple pieces of music, jointly inferring the structure within each piece (for example when sections of the score are repeated), and being able to use approximate sequential inference for longer pieces of music. Hidden Markov models have been used for audio and score alignment by Orio and Déchelle [2001], Cano et al. [1999]; and Raphael [2004] also uses a probabilistic generative model for score alignment.

8.1.2 Model Statement

The data we observe consists of N pieces of music, with T_n frames in each piece, $n = 1, \ldots, N$. Each frame of music is denoted by $y_t^{(n)}$, $t = 1, \ldots, T_n$. The score, which is the underlying representation of all of the pieces, is divided into M sections. Over each section the properties of the score are stationary, hence note onsets and offsets for example mark the beginning of new sections. In regular classical music, each bar of the score may be divided equally, for example into 32 if no notes are longer than semi-quavers. We consider the method in which the score is converted into musical audio via the concept of a 'score pointer', which denotes at time t the current position in the score where the performance is. If someone were to listen to a piece of music and follow the score with their finger at the same time, the position of the score pointer would be the position of the listener's finger. The path of the score pointer through each piece of music is denoted $x_t^{(n)}$ which takes values $1, \ldots, M$. The score pointer may move forwards and also backwards in the case of repeated sections, therefore it is not necessary in general for $x_t^{(n+1)} \geq x_t^{(n)}$.

To proceed, we require a generative signal model which assigns a probability $p\left(y_t^{(n)}|\theta_m\right) p\left(\theta_m\right)$ for all values of t, n, m. θ_m denotes a set of parameters which describes how the signal changes given the position in the score. Some of the parameters θ_m will be unknown and must be inferred as part of the audio matching task. We also define a Markov model $p\left(x_t^{(n)}|x_{t-1}^{(n)}\right)$ as a prior on the dynamics of the score pointer through each piece, with initial priors on the score position $p\left(x_1^{(n)}\right)$ and the signal $p\left(y_1^{(n)}|\theta_m\right)$. The joint probability distribution of this model is given by

$$\prod_{n=1}^{N} p\left(y_1^{(n)}|\theta_{x_1^{(n)}}\right) p\left(\theta_{x_1^{(n)}}\right) p\left(x_1^{(n)}\right) \prod_{t=2}^{T_n} p\left(y_t^{(n)}|\theta_{x_t^{(n)}}\right) p\left(\theta_{x_t^{(n)}}\right) p\left(x_t^{(n)}|x_{t-1}^{(n)}\right) \tag{8.1}$$

8.1.3 Interpretation of Dynamic Time Warping

Dynamic time warping constructs a set of M matches between the frames of two pieces of audio, in such a way that the set of matches represents the path of an unknown score pointer through both pieces. Each match m is a unique pair of frames $\left\{y_{p(m)}^{(1)}, y_{q(m)}^{(2)}\right\}$ from each piece, where $p(m) \in 1, \ldots, T_1$ is the timing of the match in piece 1, and $q(m) \in 1, \ldots, T_2$ is the timing of the match in piece 2.

The cost of each match $d\left(y_p^{(1)}, y_q^{(2)}\right)$ (also known as the *distance* or *similarity* between the frames) is a function of the two frames. The cost has a small value when the frames share some characteristics that indicate that they are at the same position in the unknown score. A good cost function is insensitive to non-score related features, such as the overall energy of the frame. The cosine distance which normalizes each vector is a popular choice.

$$d\left(y_p^{(1)}, y_q^{(2)}\right) = \frac{y_p^{(1)} \cdot y_q^{(2)}}{\left|\left|y_p^{(1)}\right|\right| \left|\left|y_q^{(2)}\right|\right|}$$

where $||y||$ is the l^2 norm.

There is also an additional cost between consecutive matches, which controls how the score pointer moves frame by frame through both pieces. This function ensures that only realistic score pointer paths are acceptable and disallows large differences in time between consecutive matches in both pieces, i.e., both $p(m) - p(m-1)$ and $q(m) - q(m-1)$ are constrained to be small for all m. Formally, we define a cost function

$$c\left(p(m-1), p(m), q(m-1), q(m)\right)$$

which is the cost of moving the score pointer from $p(m-1)$ to $p(m)$ in piece 1 and from $q(m-1)$ to $q(m)$ in piece 2. The value of the cost function must be infinite for any movement of the score pointer disallowed by the application. For example, if the application does not allow the score pointer to move backwards in time, then the cost function must be infinite for $p(m) - p(m-1) < 0$ or $q(m) - q(m-1) < 0$. For every frame in each piece to be matched with a frame in the other piece, we cannot allow the cost function to skip frames, hence it must also be infinite for $p(m) - p(m-1) > 1$ or $q(m) - q(m-1) > 1$. In the literature, the cost function is normally chosen to be stationary, in that it does not depend on the actual values of $p(m-1), p(m), q(m-1), q(m)$, but only on the differences between them, i.e.,

$$c\left(p(m-1), p(m), q(m-1), q(m)\right) = c\left(p(m-1) + t, p(m) + t, q(m-1) + t, q(m) + t\right) \forall t \in \mathbb{Z}$$

When the cost function is stationary it can therefore be written as

$$c\left(p(m-1), p(m), q(m-1), q(m)\right) \equiv c\left(p(m) - p(m-1), q(m) - q(m-1)\right)$$

Later in this chapter we will introduce a dependency on the position of onsets in the score, hence we do not simplify the specification of the cost function at this point.

A DTW algorithm minimizes the overall cost

$$\sum_{m=1}^{M} d\left(y_{p(m)}^{(1)}, y_{q(m)}^{(2)}\right) + \sum_{m=2}^{M} c\left(p\left(m-1\right), p\left(m\right), q\left(m-1\right), q\left(m\right)\right) \tag{8.2}$$

over the set of matches $\left\{p\left(m\right), q\left(m\right) : m = 1, \ldots M\right\}, M$, subject to the constraints

$$p\left(1\right) = 1, q\left(1\right) = 1, p\left(M\right) = T_1, q\left(M\right) = T_2 \tag{8.3}$$

This set of matches is known as a *path* and denotes the position of the score pointer in each piece throughout the entire path. The constraints require that the path of the score pointer starts at the first frame of both pieces, and terminates at the last frame of both pieces. Note that the length of the path M is also unknown, although a shorter length is preferred by the cost function, which normally indicates a better match between the two pieces.

The constraints may be relaxed to allow for silences or missing notes. In this case, we do not apply the constraints (8.3) but instead define *edge* costs: $e\left(p\left(1\right), q\left(1\right)\right)$ which defines the cost of starting the path at $p\left(1\right), q\left(1\right)$ and $f\left(p\left(M\right), q\left(M\right)\right)$ which determines the cost of terminating the path at $p\left(M\right), q\left(M\right)$. The overall cost is now written as:

$$e\left(p\left(1\right), q\left(1\right)\right) + f\left(p\left(M\right), q\left(M\right)\right) + \sum_{m=1}^{M} d\left(y_{p(m)}^{(1)}, y_{q(m)}^{(2)}\right) + \sum_{m=2}^{M} c\left(p\left(m-1\right), p\left(m\right), q\left(m-1\right), q\left(m\right)\right)$$

The DTW model may be interpreted using the generative model in the previous section. The cost functions are interpreted as negative log probabilities, such that the summations in (8.2) correspond to the products in (8.1), and minimizing the cost function is equivalent to maximizing the likelihood. This interpretation is powerful as the model may be extended with further prior information if available.

The cost of matching frames is equivalent to the negative log marginal likelihood of both frames being generated by the same score

$$d\left(y_{p(m)}^{(1)}, y_{q(m)}^{(2)}\right) \equiv -\log \int p\left(y_p^{(1)} | \theta_m\right) p\left(\theta_m\right) p\left(y_q^{(2)} | \theta_m\right) p\left(\theta_m\right) \, \mathrm{d}\theta_m \tag{8.4}$$

and the cost of the score pointer movement between successive matches is related to the transition probabilities:

$$c\left(p\left(m-1\right), p\left(m\right), q\left(m-1\right), q\left(m\right)\right) \equiv -\log p\left(x_{p(m)}^{(1)} | x_{p(m-1)}^{(1)}\right) - \log p\left(x_{q(m)}^{(2)} | x_{q(m-1)}^{(2)}\right) \tag{8.5}$$

The edge cost at the beginning of the path are equivalent to the priors:

$$e\left(p\left(1\right), q\left(1\right)\right) \quad = \quad -\log p\left(x_{p(1)}^{(1)}\right) p\left(x_{q(1)}^{(2)}\right)$$

and the cost at the end of the path may similarly be written as

$$f\left(p\left(M\right), q\left(M\right)\right) \quad = \quad -\log \int p\left(x_{p(M)}^{(1)} | x_{p(M-1)}^{(1)}\right) p\left(x_{q(M)}^{(2)} | x_{q(M-1)}^{(2)}\right) \, \mathrm{d}p\left(M-1\right) \mathrm{d}q(\mathrm{M}-1)$$

marginalizing over the preceding position of the score pointer.

8.2 Score Alignment

A common way to use audio alignment techniques to align a score to a piece of musical audio is to synthesize the score to audio using an electronic instrument and compute a joint path through both pieces using DTW as described in the previous section. However, the inferred path through the observed audio may not be appropriate to infer the position through the score in every frame. A single frame in the observed audio may be matched to multiple frames in the synthesized audio which may span more than one score event. Thus there is ambiguity over which score event the frame in the observed audio should be matched to.

In cases where the length of a frame is small in comparison to the minimum length of a score event, it is impossible for multiple score events to occur within the same frame of the observed audio. When a single frame is matched to multiple score events, the path inferred though the audio must therefore be incorrect. We suggest that this is due to the prior model (8.4) which matches frames being too weak, and the transition model (8.5) being too flexible, allowing dramatic changes in tempo.

In this section we focus on the development of a stronger transition model, i.e., the movement of the score pointer through the audio. We state the transition model in such a way that it can be incorporated into existing DTW cost functions and also expressed as a hidden Markov model, which allows further development of the prior models.

8.2.1 Treatment of Score Events

In score alignment, there is much more value in accurately inferring the timings of note onset events in the score rather than a smooth contour of the tempo through a piece of audio. Note onsets are also usually more important than note releases. In piano music, the sustain pedal can blur the timings of released notes, however note onsets are clearly defined. In percussive instruments, including plucked and hammered strings, the notes may not be explicitly cut off, but allowed to sound and decay. In *legato* playing, note releases are timed with the onsets of the next note, whilst in *staccato* playing, the note lengths are short and the exact position of the release may be difficult to determine in the score. Perceptually, minor errors in the timing of a note onset have a greater impact than in the timing of a release.

In light of this, we treat note onsets with a rigid prior model to attempt to infer the timings as accurately as possible. Away from the onsets however, we allow the score pointer to move flexibly through the progression of the note and its release.

8.2.2 Dynamic Time Warping Cost Function

Let the observed audio in 8.1.3 be $y_t^{(1)}$ and the synthetic audio, which is known to be aligned to the underlying score, be $y_t^{(2)}$. We know *a priori* which frames in the synthetic audio $y_t^{(2)}$ are related to note onsets in the score. If $y_{q(m)}^{(2)}$ is related to a note onset, then we set the cost function as

$$c\left(p\left(m-1\right),p\left(m\right),q\left(m-1\right),q\left(m\right)\right)=\begin{cases}0 & p\left(m\right)-p\left(m-1\right)=1 \text{ and } q\left(m\right)-q\left(m-1\right)=1\\ \infty & \text{otherwise}\end{cases} \tag{8.6}$$

(8.6) forces a movement of 1 frame in both pieces when there is an onset in the synthetic audio. If $y^{(2)}_{q(m)}$ is not related to a note onset, then we assume that the cost function is stationary and takes a value $c\left(p\left(m\right)-p\left(m-1\right),q\left(m\right)-q\left(m-1\right)\right)$. When the modified cost function described in this section is applied to score alignment using DTW, the constraint ensures that every note onset is uniquely identifiable in the path of the score pointer through the observed audio.

8.2.3 Hidden Markov Model Formulation

The hidden Markov model (HMM) formulation of score alignment is more powerful as it uses an explicit model of the score itself. This allows a generative model $p\left(y_t|\theta_m\right)$ of a frame of audio y_t to be used to infer unknown parameters θ_m using all of the frames in both the synthesized and observed audio corresponding to a certain set of notes in the score, and even using other frames sharing one or more notes. A library of training data may also be used as priors for the signal model. The parameter set θ_m at score position m includes the pitches and volumes of the set of notes currently sounding, plus any unknown parameters of the generative model for the frame. Any reasonable generative model for a frame of audio given the playing notes may be used, including the Bayesian models developed in the preceding chapters of this thesis.

The joint probability distribution of the frame y_t and the parameters θ_m may be written as

$$p\left(y_t,\theta_m\right)=\frac{p\left(y_t|\theta_m\right)}{p\left(y_t\right)}p\left(\theta_m\right)$$

Now if we have previously obtained several frames of music $y^{(n)}_t, t=1,\ldots,T_n$, which we denote $y^{(n)}_{1:T_n}$, through synthesis or other means which we know was generated from a score with parameters θ_m, then we can update the prior generative model with the additional data. As the below expression shows, this is equivalent to replacing the prior $p\left(\theta_m\right)$ with the posterior under the previously observed frames $p\left(\theta_m|y^{(n)}_{1:T_n}\right)$:

$$p\left(y_t,\theta_m|y^{(n)}_{1:T_n}\right)=\frac{p\left(y_t|\theta_m\right)}{p\left(y_t\right)}p\left(\theta_m|y^{(n)}_{1:T_n}\right)$$

If the prior is a conjugate prior of the likelihood function $p\left(y_t|\theta_m\right)$ then the posterior $p\left(\theta_m|y^{(n)}_{1:T_n}\right)$ is of the same family as the prior, the parameters of which are calculated using standard update rules.

In the HMM formulation, we are only interested in inferring the path of the score pointer x_t through the observed audio y_t, as the path through any synthesized audio is already known. The probabilistic model for the movement of the score pointer is simple. Usually from one frame to the next, the score pointer may either stay in the same score position or move to the next position.

$$p\left(x_{t+1}|x_t\right)=\begin{cases}p_m & x_{t+1}=m+1, x_t=m\\ 1-p_m & x_{t+1}=x_t=m\\ 0 & \text{otherwise}\end{cases} \tag{8.7}$$

p_m is related to the current tempo and the expected duration of the event represented by the score pointer position at that point. For example, if the event is expected to last 1 second and the time difference between frames is 125ms then p_m should be set to 1/8.

(8.7) may be extended to allow other transitions, such as skipping a score event in a live and error prone performance. An additional Markov model allowing changes in tempo may be added by extending the hidden state x_t to include the unknown tempo parameter. The tempo should only be allowed to change slowly, for example at bar lines in the score or explicitly marked tempo change points.

When θ_m represents a note onset, we set $p_m = 0$: similar to the constraint (8.6) to improve the accuracy of onset timings. This forces the score pointer to move one frame at the onset. The remainder of the note is represented by θ_{m+1} and may be further divided into sustain and release portions of the note if this is available or may be inferred from the score.

8.2.4 Inference

In Peeling et al. [2007a] we described two methods of inference using a simpler version of the hidden Markov model described in 8.2.3. In this section, we extend these methods into an iterative scheme which improves the accuracy of the timings and the consistency of the inferred score parameters. This scheme firstly infers $x_{1:T}$ given $\theta_{1:M}$ and then infers $\theta_{1:M}$ given $x_{1:T}$.

The method of calculating the posterior distribution $\prod_{t=1}^{T} p\left(x_t|y_{1:T}, \theta_{1:M}\right)$ for all t is known as the *forwards-backwards* algorithm [Rabiner, 1989]. Modifications of this algorithm be used on-line to calculate a fixed lag filtering distribution $p\left(x_t|y_{1:t+L}, \theta_{1:M}\right)$ where $L \geq 0$ is the permitted lag allowed in observations before the score pointer position is required.

The method of calculating the mode of the posterior distribution is known as the Viterbi algorithm. It is used where a consistent path of the score pointer is required across the entire piece.

The remaining step of inference is to update the parameters θ_m of the generative model, given that we have attempted to fit the score to the observed frames. The standard method for hidden Markov models is the Baum-Welch algorithm, an expectation-maximization algorithm. The algorithm first computes the posterior distribution of the score pointer by the forward-backwards procedure, and then locates the MAP estimate of $\theta_{1:M}$ under the posterior distribution of the parameters

$$p\left(\theta_{1:M}|x_{1:T}, y_{1:T}\right)$$

An alternative method, known as conditional modes, uses the Viterbi path to compute the score pointer, and then maximize

$$p\left(\theta_m|\left\{x_t, y_t\right\} : x_t = m\right)$$

for all m. As this method segments the score parameters and data into separate maximization problems, there is potential for the computation to be carried out in parallel; and in general the computation and memory requirements are less than employing the Baum-Welch algorithm.

8.2.5 Results

Figure 8.1 on page 144 shows an example of audio alignment carried out using DTW techniques on a recorded piece of guitar audio aligned to a synthesized version. The synthesized version was generated from the known score of the piece with a constant tempo. The note onsets were identified from the score, and timings were assigned by scaling the number of beats from the beginning of the score to the note onset event. Note onset

costs are implemented as described in 8.2.2. The distance matrix $d(p,q)$ is computed using a single source Gaussian variance model (Section 7.2) as follows. Each element of the distance matrix is the joint marginal likelihood of the frame of recorded audio $y_p^{(1)}$ and the frame of synthetic audio $y_q^{(2)}$, assuming that both frames share the same template vector $\mathbf{t}$ but have a separate excitation parameter: $v_p^{(1)}$ being the excitation parameter for $y_p^{(1)}$ and $v_q^{(2)}$ being the excitation parameter for $y_q^{(2)}$. We therefore calculate

$$\begin{aligned} d\left(y_p^{(1)}, y_q^{(2)}\right) &= p\left(y_p^{(1)}, y_q^{(2)}\right) \\ &= \int p\left(y_p^{(1)}|\mathbf{t}, v_p^{(1)}\right) p\left(y_q^{(2)}|\mathbf{t}, v_q^{(2)}\right) p(\mathbf{t})\, p\left(v_p^{(1)}\right) p\left(v_q^{(2)}\right) \mathrm{d}\mathbf{t}\mathrm{d}v_p^{(1)}\mathrm{d}v_q^{(2)} \end{aligned} \tag{8.8}$$

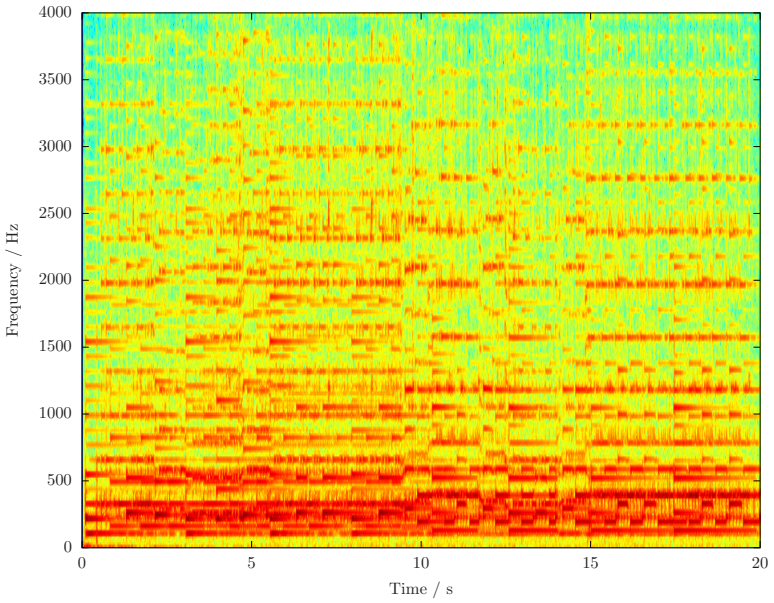
for each pair of frames, using the Variational Bayes algorithm 7.1 to approximate the integral in (8.8). The template and excitation hyperparameters are chosen to be uniform: $a^t = 2, b^t = 1$ and $a^v = 2, b^v = 1$, and the hyperparameters are not optimized for this algorithm.

The alignment path is then computed by DTW. The result is that the timings of note onsets in the recorded piece are synchronized well with the synthetic version, which is indicated by the alignment path passing through the intersections of the edges of the vertical and horizontal bands in the distance matrix. Figure 8.1a on page 144 shows the spectrogram of the recorded audio, and Figure 8.1b on page 144 shows the distance matrix computed from (8.8) overlaid with the alignment path. By comparing with the distance matrix with the spectrogram, it can be seen that the note onsets of the observed audio occur at the edges of the vertical bands. Similarly, note onsets in the synthetic audio correspond to the edges of the horizontal bands. On close observation, the alignment path passes through the intersections of the band edges, which indicates that the note onsets in the observed and synthetic audio have been matched together with little timing error.

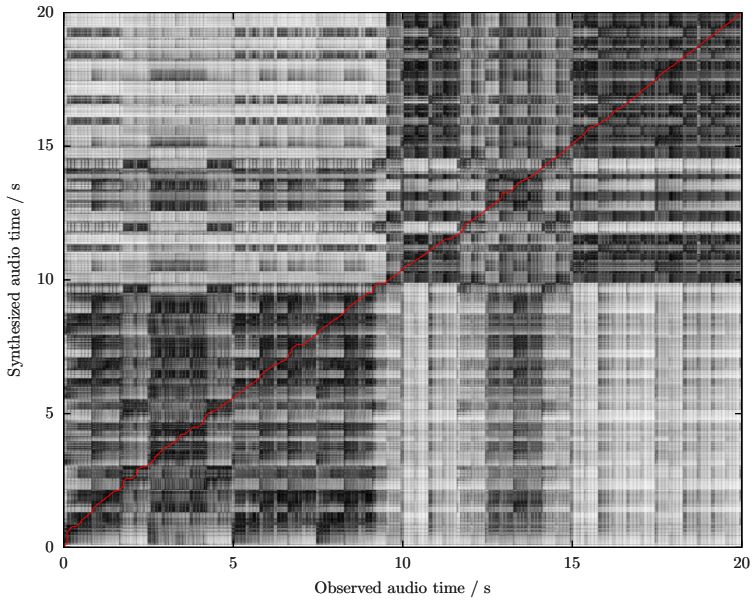
To quantify the improvement in the alignment when using note onset costs and the iterative inference algorithm of 8.2.4 we use a data set built from MIDI and MP3 files from the Classical Piano MIDI page[1]. The MP3 audio files are recorded acoustically from a MIDI controlled grand piano, and are aligned to the MIDI files provided. An accompanying synthetic set of audio was obtained by removing all of the tempo and expressive markings from the MIDI files and synthesizing the result. Both pieces of audio were downsampled to 8000Hz and split into frames of 48ms with 50% overlap between the frames. These were then matched together using the algorithms described in this chapter. The tempo through the observed audio is assumed to be constant, and is estimated from the tempo of the synthetic audio by multiplying by the ratio of the lengths of the two pieces. The estimated tempo is then used to set the cost function for moving from one frame to the next using the model in (8.7).

For each note onset in the score, we identify the frame in the synthetic audio containing the note onset. Then if that frame is matched to a unique frame in the observed audio by the DTW algorithm (which is guaranteed if the note onset costs described in 8.2.2 are used), then the centre of the frame in the observed audio is recorded as the timing of the note onset in the observed audio. If the frame in the synthetic audio is not matched to a unique frame in the observed audio, then the timing of the onset in the observed audio is recorded as halfway between the start of the first frame and the end of the last frame of the group of frames in the observed audio matched to the frame containing the onset in the synthetic audio.

[1] `www.piano-midi.de`



(a) Spectrogram of the observed audio. Vertical features in the spectrogram correspond to vertical bands in the distance matrix below.



(b) The distance matrix of the observed spectrogram above, compared frame-by-frame to synthesized audio from the same score.

Figure 8.1: On the distance matrix of the observed spectrogram, regions of high spectral similarity are shaded darker and regions of low spectral similarity are shaded lighter. Overlaid in red is the optimal alignment path computed using DTW, which moves steadily from the beginning of both pieces along the diagonal of the distance matrix.

		Cosine Distance		Gaussian Variance		
Piece	Unaligned	DTW	Note Onset Costs	DTW	Note Onset Costs	Iterative Inference
alb_esp1	578.1	350.1	342.7	357.8	345.3	331.8
scn15_5	1278.5	11.9	11.6	11.3	10.9	10.6
bor_ps7	822.0	55.4	51.5	50.6	47.0	45.8
alb_esp2	1203.4	122.7	119.9	111.5	110.0	110.0
mendel_op62_4	607.5	16.0	15.9	15.8	15.4	14.7
ty_maerz	3068.1	639.0	634.1	637.6	633.6	625.7
chpn-p22	753.2	158.4	152.7	154.9	151.1	151.0
scn15_3	74.2	12.4	12.2	17.0	13.4	12.4
scn15_1	374.4	13.2	12.5	12.6	12.2	12.2
scn15_6	408.0	27.2	22.8	22.7	22.2	19.4

Table 8.1: Score alignment: median alignment in milliseconds

We then calculate the difference in milliseconds between the onset timings in the original MIDI (before removing tempo changes) compared to the note onsets identified in the aligned audio. The quality of alignment measured using the median alignment error of note onsets in milliseconds, as this evaluation criterion is also used in Cont et al. [2007], Devaney et al. [2009] for score alignment. The results are presented in Table 8.1 on page 145.

From these results we observe that both the cosine distance and Gaussian variance models produce similar alignments. The advantage of being able to use an iterative inference scheme for the Gaussian variance model provides modest improvements in alignment accuracy. The accuracy of the alignment varies strongly based on the piece that is being aligned. The common characterstic of the pieces which are badly aligned is that they include sections of fast notes with high polyphony, and in these situations a simple distance metric such as the cosine distance or a single source Gaussian variance model is too weak a model to improve alignment dramatically. We therefore suggest that a useful line of future work would be to investigate more powerful models, which in themselves are capable of music transcription, such as the multiple source Gaussian variance model in Chapter 7 with training data in addition to the synthetic audio, and implement them in the inference framework described in this chapter, in order to tackle these more difficult cases of score alignment.

In Figure 8.2 on page 146 we show how the spectrogram audio and aligned score can be presented in an visual manner. The score pointer is presented as a vertical bar which moves through both the spectrogram and the score, synchronized with the audio track.

8.3 Event Based Inference

The generative model described in 8.1.2 directly works with frames of observed audio, and infers the temporal structure from the posterior distribution. In this section, we consider a different application, where the observations are not frames of audio but incompletely labeled events. The observations here are the output of an audio preprocessing system such as an onset detector (which detects rapid changes in the energy of the audio over time), or an approximate transcription of the melodic line of the piece, which could be provided by a human in the context of a query-by-humming application. In these situations, we again wish to determine

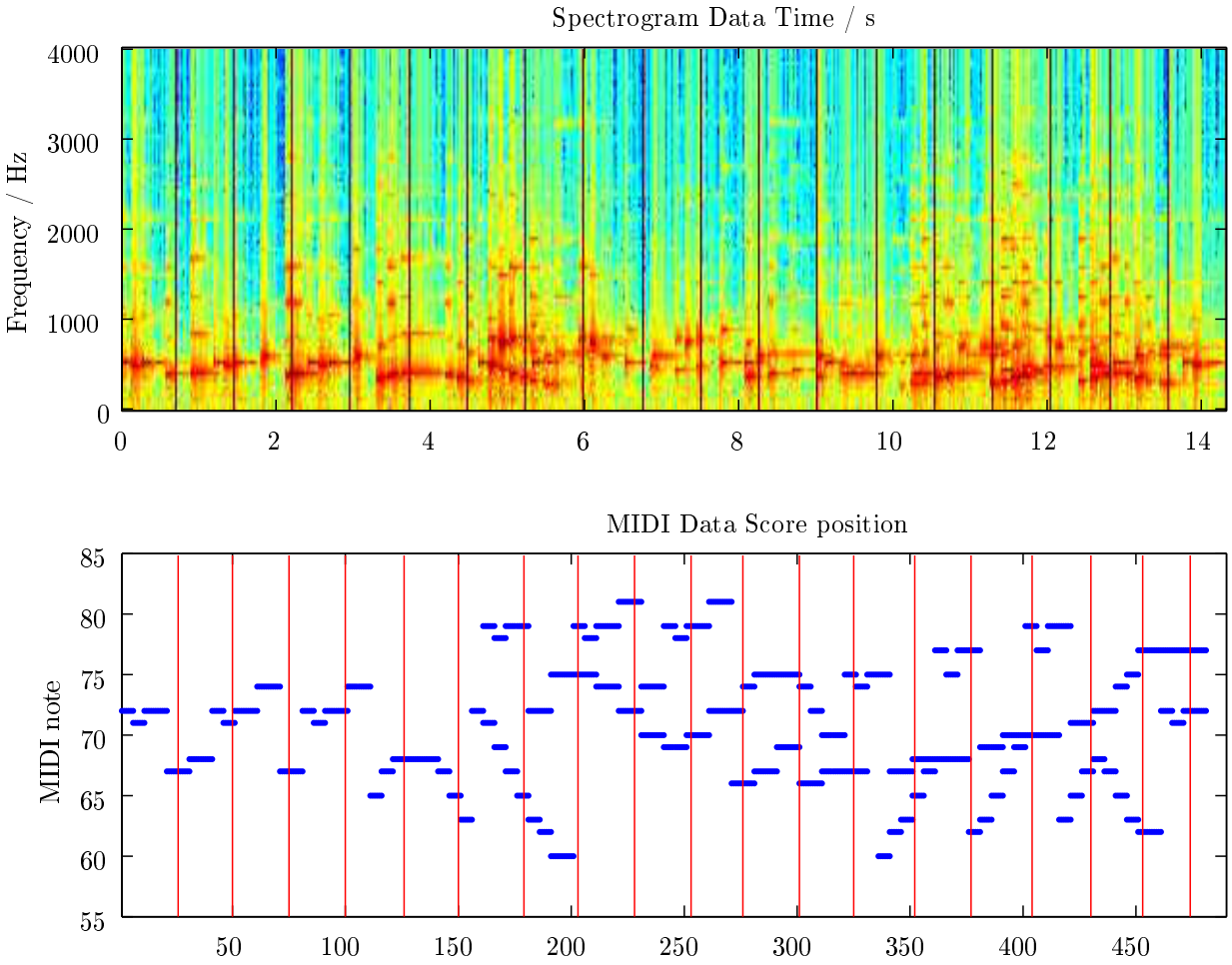


Figure 8.2: Score alignment using Gaussian variance model. The movement of the score pointer is displayed regularly in time as a vertical bar passing through the spectrogram and also the MIDI representation. The spectral features can be matched visually to the score representation easily, and it can be seen that the score pointer positions shown correspond to the appropriate timing in the audio. In a practical visualization application, this figure can be presented as a video, where the score pointer moves through the spectrogram and the MIDI representation concurrently, whilst the audio is playing. It is then much easier to notice subtle changes in tempo (in performance rather than an explicit score marking), for example between score positions 150 and 175, which marks the entry of the second voice in Bach's second Fugue in C minor (from the Well-Tempered Klavier performed by Daniel Ben Pienaar)

the temporal structure in the music, and be able to match and align the observed events with the underlying score. The goal of this section is the same as in Section 8.2, however instead of using a generative model of the audio in each frame, we now must define and infer using a model which describes the production of these observed events from the score.

For example, a tempo tracking algorithm must firstly infer the presence of onsets of note events in a piece of music, and then infer the overall speed of the arrival of events by looking at periodic patterns in the timing of events (Figure 8.3 on page 151). The onset timings may also be provided directly by a human listener, in an application which infers and even controls the tempo of a piece playing, or in a query-by-tapping context, where the application attempts to match the timings provided by the listener to a MIDI database. In these applications, the events are unlabeled - we do not know which beat of which bar each event belongs to.

Exact onset timings may also be acquired from a MIDI enabled instrument attempting to perform a score. Analyzing the note timings in relation to the score is useful for musicological studies of performance. Another application is an incomplete transcription of the score, such as a melodic line transcription in a query-by-singing application, or even the output of a classification model-based transcription system which cannot easily be set in a generative model framework. In these applications, the events are partially labeled: we know the pitch and perhaps the volume of the note, but these notes are not matched with the score itself. Some of the notes observed may be erroneous, and others may arrive in the wrong order, even subtly in the case of piano chords and polyphonic music.

In the above descriptions, we have motivated both the use of an alignment application (to infer the tempo throughout the audio) and a matching application where we wish to select the best match to the observed events from a large set of candidate pieces. Although these applications appear different, they can both be addressed by Bayesian inference of the unknown model parameters. For alignment, inference of the note onset positions can be used to construct the progress of the tempo through the piece. For matching against other candidates, Bayesian inference can compute the likelihood of the observed events given a candidate score. The candidate giving the highest likelihood to the observations is therefore the best match to the observations. In 8.3.3 we consider a query-by-tapping application, where we attempt to match a set of onsets intended to mimic the rhythmic structure of a piece of music, to candidate scores from a database.

In terms of the generative model $p\left(y_t|\theta_m\right)p\left(\theta_m\right)$, y_t refers to the set of event onsets inferred in that frame. In this section we will propose a Bayesian model which can be adapted to all of the above applications in a general way. We define C as the number of categories of events that we are able to observe. If an event of category $c \in 1,\ldots,C$ occurs in frame y_t, then we use the notation that $c \in y_t$, and if it is expected to occur during the section of score θ_m then $c \in \theta_m$.

The definition of an event category depends on the nature of the observed events. If the observed events are generated by a simple onset detector, there is only one event category: the detected onsets. Simultaneously sounding note onsets are grouped into score onset sections. In a score onset section $c \in \theta_m$ as we expect an onset to be detected when there is an onset in the score. Sections are also defined for the periods in the score where there are no onsets. In these sections, $c \notin \theta_m$ as we do not expect an onset to be detected.

If the observed events are note onsets returned by a melodic line transcription algorithm, then each pitch returned by a transcription algorithm would correspond to one of the C categories. The score is divided into disjoint sections, where a group of pitches are sounding throughout a section, or there is silence. When a

pitch corresponding to event category c is sounding in section m then $c \in \theta_m$ as we expect the transcription algorithm to detect this pitch during this section of the score. In the case of silence, we expect $c \notin \theta_m$ for all c.

8.3.1 Counting of Temporal Events

The basic model we propose maintains a count of the number and category of temporal events observed in the music up to and including the present frame. Our observation is a function $n_c(t)$ for every frame y_t defined as the number of events of category c observed up to and including frame y_t.

Our prior model consists of the expected number of events observed at different times in the piece of music. We will model the temporal events as a set of independent non-homogeneous Poisson processes, one for each category. The occurrence of note c onsets has a time-varying intensity $\rho_c(t)$ which for each value of t gives the expected number of onsets by the time we reach frame t, i.e.,

$$n_c(t) \quad \sim \quad \mathcal{P}o\left(\sum_{\tau=0}^{t} \rho_c(\tau)\right) = \mathcal{P}o\left(\lambda_c(t)\right) \tag{8.9}$$

where $\lambda_c(t) = \sum_{\tau=0}^{t} \rho_c(\tau)$.

The intensity function $\rho_c(t)$ gives the expected number of event of category c occurring in each frame. When matching onset detections to a score, we would initially expect that this intensity function is equal to the number of events in the section of the score x_t corresponding to the frame t, i.e.,

$$\rho_c(t)\,|\theta_{x_t} = \begin{cases} 1 & c \in \theta_{x_t} \\ 0 & c \notin \theta_{x_t} \end{cases} \tag{8.10}$$

This model of the intensity function is suitable for ideal cases where we do not expect any errors in the performance or errors in the detection of the events, and the only unknown variable is the tempo of the performance, represented by x_t which maps frame t to score section m.

(8.9) is a generative model for the occurrence of observed events, given the expected counts of events in the underlying score, and therefore may be used for all of the applications described earlier. Alignment applications involve inferring the tempo of the performance x_t throughout the piece. Applications which match the observations to the best candidate score must compute the likelihood of the observed event counts $n_c(t)$ given the candidate score $\theta_{1:M}$

$$\prod_{c \in C} \prod_{t=1,\ldots T} p\left(n_c(t)\,|\theta_{1:M}\right) \tag{8.11}$$

for each candidate. The candidate with the highest likelihood (8.11) is chosen as the best match.

The Poisson assumption allows variability in both the number of events detected and their timings. The maximum likelihood case is when the events match the score exactly, however we show in 8.3.3 that using (8.10) without modification is sufficient to match scores to observations on a global or coarse scale.

8.3.2 Clutter and Missed Detections

For a more powerful model which is more appropriate for real performances and onset detection methods and is able to match individual events in the score to the observations, we may begin by adding two additional error parameters per event type. The first parameter is a *clutter* process $\rho_c^{(\text{clutter})}$ which gives the expected number of spurious event detections in each frame. The second parameter governs the probability of missing note detections, which we denote $\rho_c^{(\text{missed})}$. The new model for the intensity function is

$$\rho_c\left(t\right)|\theta_{x_t} = \begin{cases} 1 - \rho_c^{(\text{missed})} & c \in \theta_{x_t} \\ \rho_c^{(\text{clutter})} & c \notin \theta_{x_t} \end{cases}$$

In this section, we treat the two error processes as independent per event type and constant throughout the score / observation. It is straightforward to allow the error parameters to vary across different parts of the score, for example in fast moving sections where notes are more likely to be missed, by appropriately subdividing the score.

These parameters are straightforward to infer as part of the iterative procedure described in 8.2.4. If the current MAP estimate of the path of the score pointer is given by x_t^* for all t, then the maximum likelihood estimate of the missed event parameter is

$$\rho_c^{(\text{missed})} = \frac{1}{T}\sum_{t=0}^{T} \mathbb{I}\left[c \in \theta_{x_t^*}\right] \mathbb{I}\left[c \notin y_t\right]$$

is the average of the missed detections in the observation, and the maximum likelihood estimate of the clutter parameter is

$$\rho_c^{(\text{clutter})} = \frac{1}{T}\sum_{t=0}^{T} \mathbb{I}\left[c \notin \theta_{x_t^*}\right] \mathbb{I}\left[c \in y_t\right]$$

is the average of the spurious detections.

If we are repeatedly using a particular technique for detecting events, it is likely that we would have strong prior information about the clutter and missed event processes. In this case, we can put a Beta distribution (Section A.4) $\rho_c^{(\text{missed})} \sim \mathcal{B}\left(\alpha_c^{(missed)}, \beta_c^{(missed)}\right)$ on the parameter, where $\alpha_c^{(missed)} + \beta_c^{(missed)}$ is the number of times we applied the technique and $\beta_c^{(missed)}$ is the number of missed detections. The maximum a posteriori estimate of the missed event parameter is

$$\rho_c^{(\text{missed})} = \frac{1}{T + \alpha_c^{(missed)} + \beta_c^{(missed)}} \left(\beta_c^{(missed)} + \sum_{t=0}^{T} \mathbb{I}\left[c \in \theta_{x_t^*}\right] \mathbb{I}\left[c \notin y_t\right]\right)$$

An equivalent formula holds for the clutter parameter

8.3.3 Query-by-Tapping Results

The MIREX Query by Tapping Task is an interesting setting for evaluating models of temporal structure in music. A set of onset times are provided to the system, and the task is to match the rhythmic structure observed to one of a set of candidate scores. The MIREX task provides a set of monophonic MIDI database

records, and a set of audio tapped and symbolic queries available for download[2]. The question here is whether the temporal structure of music can be adequately represented by rhythmic (onset) information alone [Jang et al., 2001]. This has implications for score-alignment algorithms where robustness may be increased by ignoring pitch information rather than including it.

Applying the models here is straightforward. $C = 1$ and $n_c(t)$ is the number of observed onsets provided in the query up to frame t. $\rho_c(t)$ is set to the number of note onsets in the MIDI file up to frame t. The examples in the MIREX task are monophonic, so note onsets do not happen at exactly the same time.

The system ranks all of the MIDI files in the database according to the likelihood (8.11) for each query. Figure 8.3 on page 151 shows an example of the inter-onset timings which are used to match the observed onsets to the note onsets in MIDI. To evaluate how well the system performs, for each query q, we compute the rank r_q of the correct score when the database scores are ranked according to likelihood. If the system correctly matches query q by assigning the highest likelihood, then $r_q = 1$. If the system assigned two incorrect scores with a higher likelihood than the correct score, then $r_q = 3$. The evaluation method published by MIREX is the mean reciprocal rank (MRR) over all the queries $q \in Q$:

$$\text{MRR} = \sum_{q \in Q} \frac{1}{r_q}$$

such that a perfect system which returns the correct score for every query, i.e., $r_q = 1 \forall q$ has an MRR of 1, and systems which make more mistakes have lower MRR values.

. The best performing algorithm by Typke and Walczak-Typke [2008] for the 2008 task, which calculates an Earth mover's distance (EMD) between the observed and score onsets, achieved an average MRR of 0.52. Even without using any Bayesian techniques, we obtain an average MRR of 0.54 on the MIREX database, outperforming more elaborate and computationally expensive algorithms on the symbolic onset data.

8.4 Conclusion

In this chapter we have developed techniques for applying a generative model of a musical audio signal coupled with a Markov model of the movement of a score pointer to a variety of inference tasks in musical signal processing. The first application we consider is aligning two extracts of musical audio by matching frames on the basis of the similarity of spectral features. A popular framework for carrying out this task is dynamic time warping (DTW). By expressing the dynamic time warping problem in a generative model setting, we are able to incorporate Bayesian priors on the spectral features and the matching process in order to generate more reliable and realistic results. Moreover we are able to extend the method to multiple extracts and apply iterative Bayesian inference techniques on hidden Markov models (8.2.4) in order to process frames in real time and over indefinitely long periods of time, which is not possible with a standard DTW implementation.

The second application we consider is score alignment, where we typically have a symbolic representation of a piece of music which is to be matched to an extract of audio. The audio is expected to be a performance of that score with tempo changes and some errors. We sketch a model of the score pointer which assigns more importance to the position of onsets in the score, and allows gradual tempo changes. We also show

[2] www.music-ir.org/mirex/wiki/2008:Query_by_Tapping

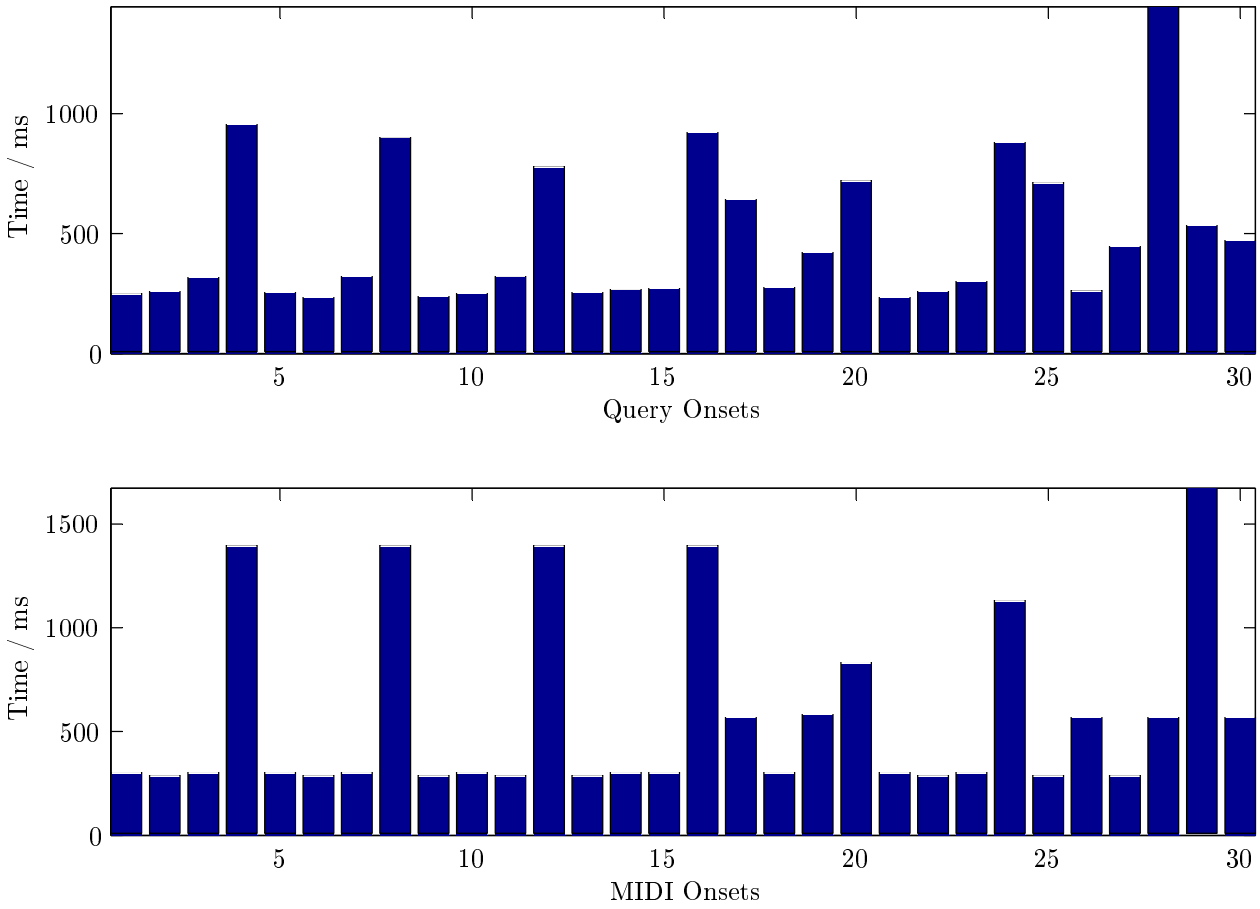


Figure 8.3: Inter-onset timings in a query-by-tapping problem. The timings in the tapped query have the same rhythmic structure as the MIDI timings, which enables the query to be matched with a high likelihood to the score using the Poisson process model generated in this chapter.

how training data, for example a synthesized version of the score, can be used to infer the parameters of the generative model, and how to update the model using the structure inferred within the observed audio itself. A standard approach to score alignment is to match a synthesized track with the observed audio using DTW. By applying the prior model of the score pointer position and the improved inference techniques, we show how the accuracy of the matching can be improved, especially at the position of note onsets where timing errors are perceptually most critical.

The final applications are based on an event-based observation of the audio, which may be obtained through an onset detector or a principal pitch transcription algorithm. We develop a novel event counting generative model of the events using a non-homogeneous Poisson process, which replaces a generative model of the signal itself, and may use the same inference techniques as the other models in this chapter. We describe simple priors for the event processes which allow and infer the probability of missed event detections and clutter for each event type. The approach is applied to a query-by-tapping example where it is shown to be more accurate than more elaborate algorithms.

The work of Degara et al. [2010] shows that significant improvement in note onset detection can be achieved by applying a prior model of rhythmic structure and fusing onset detection with rhythmic structure constraints. A natural extension of the work in Section 8.3 would be to jointly infer the event positions and the score structure using Bayesian inference.

Throughout this chapter we have mostly outlined the prior models and inference algorithms that can be applied using this framework, whereas in previous work [Peeling et al., 2007a] we have expressed a specific approach in more detail. The hidden Markov model framework is general and well studied, and many software packages exist for inference in HMMs which only require the specification of the observation model and the dynamics of the hidden state. Our contribution in this chapter has been to illustrate how several important applications in musical signal processing may be carried out through these inference algorithms.

Chapter 9

Conclusion

9.1 Summary

In this thesis we have presented a variety of Bayesian models and modelling methods aimed at tackling difficult applications of the processing of musical audio signals. A hierarchy of Bayesian priors is a feasible way to represent the complex structure exhibited in musical audio that is known from musical theory and has been obtained by experiments on the physics of musical instruments. Generative models of music are attractive as they are not tied to a particular application. Instead, the same generative model and prior structure may be used for music transcription, source separation, synthesis and reconstruction, simply by identifying which parameters of the model are known and unknown in a particular context, and applying an appropriate inference algorithm to the model to infer the values of the unknown parameters.

For each model we have developed in this thesis, we have described how it differs from existing work or generalizes an existing model, and provide a theoretical basis and justification to the accuracy observed in the experimental results presented. We have focused particularly on the problem of multiple pitch detection in frames of audio, particularly because it is an exacting measure of how appropriate and accurate a model of a musical signal is, especially for mixtures of three or more notes. Our assumption is that a generative model which is capable of performing the difficult classification task of identifying multiple pitches with high accuracy will also produce faithful and realistic reconstructions when used for its original purpose in a synthesis application, as far as the limitations of the particular model allow[1].

Another goal of this thesis was to produce algorithms and inference methods using these generative models that are ultimately feasible for real-time applications. Musical signals are inherently complex, and even a simple model requires many parameters to model a single frame of audio. Full Bayesian inference by reversible jump MCMC when the number of parameters is itself unknown is slow despite substantial improvements to inference in the Bayesian harmonic model in previous work. It is important however that any compromises made should be with the inference algorithm rather than by oversimplifying the model, because musical audio requires a rich prior structure to capture the information we are extracting from the signal.

[1] The matrix factorization models in Chapter 7 have no explicit model for the phase of the source coefficients, hence the phase of a synthetic signal would need to be constructed using a phase vocoder [Flanagan et al., 1965, Cemgil and Godsill, 2005].

In Chapter 5 we embellished the existing Bayesian harmonic model with a justification for using the Hilbert transform to improve the models ability to capture frequency and amplitude modulations in a partial frequency. We also saw a small improvement when using sinc windowed Gabor basis functions to model signals from instruments with vibrato. We also provided a result for the mode of the posterior distribution of the signal-to-noise ratio parameter which reduces the number of parameters that need to be simulated in the MCMC algorithm. By making these modifications to the existing model, we were able to show improvement in the accuracy of multiple pitch detection for two-note mixtures, although there was no improvement demonstrated for mixtures of three or more notes, and we claimed this was due to not estimating the frequency values to sufficient accuracy. In Chapter 6 however, we make this model practical for applications demanding faster computation, by splitting inference into two stages: the first stage to detect partial frequency positions in a frame using the generative model with a vague prior over possible frequency positions, using numerical techniques to improve the accuracy of the frequency estimates; and the second stage to fit a harmonic model to the estimated partial frequencies. Both stages were implemented using greedy algorithms, but we showed that the generative signal model and the prior harmonic model are powerful enough to detect the number and frequencies of pitches in a frame to the same level of accuracy as a full Bayesian inference scheme, but with computation reduced dramatically.

In Chapter 7 each frame of audio is modelled by projection onto a fixed basis, rather than inferring the number and frequencies of the bases. The harmonic spectrum of a note is represented as a prior on the relative amplitudes of the basis functions in each frame, and the amplitude envelope of a note is represented as a prior on the relative energy of the signal across frames. The model is linear in the amplitude parameters, allowing multiple notes to be superimposed, and importantly multiple frames can be processed in parallel. A simple polyphonic transcription algorithm was implemented for this model, and was shown to have a competitive level of accuracy to other transcription schemes on a large set of classical music.

In Chapter 8 we present a unified framework for musical signal processing applications which interact with a score of the music. This framework defines the dynamics of a virtual score pointer, such that the generative model of the signal in each frame depends on the properties of the score at the position indicated by the score pointer. By treating the dynamics and the generative model together as a hidden Markov model, we have a large and flexible class of inference algorithms available to estimate the movement of the score pointer through the signal and to iteratively train and learn the parameters of the generative models on past and currently observed data.

9.2 Discussion

The use of Bayesian methods defines a clear separation between the model and the inference technique used to infer unknown quantities about that model. The use of hierarchical priors allows different models to be substituted in place of one another. For example, in Chapter 8 any of the generative signal models developed in this thesis, or elsewhere, can be introduced without modifying the overall structure of the inference algorithm, although the complexity and number of parameters of the signal model may prohibit this. In Chapter 5 we were able to substitute new models developed in the chapter into the reversible jump MCMC algorithm developed in existing work without modifying any of the parameters used in the inference. This allowed us to objectively measure and demonstrate different levels of accuracy achieved with different

model configurations.

Through the use of analogies between models, we are able to apply the techniques developed by researchers in different fields and contribute likewise. This is illustrated in Chapter 7 and prior work, where the technique of non-negative matrix factorization (NMF) has applications in document classification [Xu et al., 2003], face recognition [Guillamet and Vitrià, 2002] and chemometrics [Paatero, 1997] for example, as well as polyphonic music transcription [Smaragdis and Brown, 2003]. One major advantage of this is that any inference algorithm designed for the general statement of a problem can be used for all the applications that have been transformed into this model. For popular models there typically exist multiple approaches which differ in terms of accuracy, computation, memory, etc. In Chapter 7 we only considered inference techniques which update the two matrices separately, a technique commonly known as multiplicative update rules [Lin, 2007a]. However other authors have applied gradient descent techniques, such asLin [2007b], to reduce the number of iterations required to reach a local optima. Another example of this is constructing a Bayesian network using conjugate priors. Once the model is designed and specified, then either Gibb's sampler, an MCMC technique, or Variational Bayes can be applied, using the same expressions for the conditional distributions of the parameters[2].

Many of the models described make use of linearity in the parameters so that independent processes can be superimposed and inferred from the observation. In music, this is a good approximation, as notes on a musical instrument are mostly independent of any other notes played on that instrument and notes played on other instruments. In Chapter 6 and Chapter 7 we therefore considered the model for a single harmonic source first, and then showed how multiple sources could be superimposed with an additional source modelling background noise. In Chapter 5 we take this further, and begin with the model for a single partial frequency within a harmonic. These models may then be used for source separation, where individual components of a signal are extracted and synthesized.

9.3 Further Research

9.3.1 Improvements to the Gaussian Variance Model

Further work is needed to investigate and strengthen the relationship between the Bayesian harmonic model developed in Chapter 5 and the Gaussian variance model of spectrogram coefficients in Chapter 7. The goal of this work would be to increase the accuracy of the polyphonic transcription algorithm in Chapter 7 to that of the frame-based multiple pitch detection algorithm that was developed in Chapter 6. Both algorithms function by greedily adding notes to the transcription, so accuracy could be improved by focusing on the following areas:

1. The choice of basis functions used for the Gaussian variance model. The implementation presented in this thesis used the short-time Fourier transform to obtain the observed signal coefficient, thus assuming that each component of the signal in a frame is a sinusoid with constant amplitude. Davy and Godsill [2003] previously showed that a musical signal could be better modelled using Gabor basis functions which allows a slowing varying amplitude thoughout the frame, compared to the model of

[2]Implementing generic Gibb's sampler algorithms is provided in software frameworks such as BUGS (Bayesian Inference Using Gibbs Sampler) available at `www.openbugs.info/w/`. A similar framework exists for Variational Bayes, known as VIBES (Variational Inference for Bayesian Networks) and is available at `vibes.sourceforge.net`

Walmsley et al. [1999] which has only one amplitude parameter per frame. We also showed in Chapter 5 that using the Hilbert transform and a sinc window Gabor basis improves the accuracy of modelling. Although the frequencies of the basis function are fixed in the Gaussian variance model, the number of basis functions per frame and the shape of the Gabor windows, could be modified in order to model the signal better and obtain improvements in transcription accuracy. The basis frequencies could also be chosen to match what would be expected in a harmonic musical signal rather than being spaced equally on the frequency axis.

2. The number of template functions used to model each pitch. In the transcription algorithm only one template function was used per pitch to keep computation at a minimum, although 7.4.1 indicates that multiple templates is preferred by Bayesian model selection.

3. Deriving the relationship between the priors in the two models. As a starting point, data generated by the harmonic model could be used to train the template functions of the Gaussian variance model, so that the model priors are equivalent. However, deriving even an approximate mathematical relationship may provide more insight into the models, for example the number of templates that should be used.

9.3.2 Frame Boundaries

The algorithms in this thesis model continuity between adjacent frames only at a high level, by modelling the transition probabilities of note pitches and volumes across frame boundaries. The generative model for each frame is not directly dependent on the signal in the previous frame. This allows for potential phase discontinuities at frame boundaries, the results of which are unpleasant artifacts when a signal is reconstructed. Also, the frames of the audio obtained often overlap by 50% of the samples, in which case the frames are more strongly dependent on one another than if there was no overlap.

To improve the modelling of phase boundaries, we need to account for the fact that a basis function in the model can contribute to two adjacent frames of audio. In the case of 50% overlap, every basis function contributes to two frames, whereas when there is no overlap, only the basis functions whose region of support extends beyond the end of one frame also contributes to the signal in the next frame. One method of treating shared basis functions in an iterative multiple frame processing algorithm is to fix the parameters and amplitudes of the basis functions in one frame to the values found when they were inferred in the adjacent frame, and subsequently alternating in which frames the parameters are fixed and in which frames they are inferred. This is straightforward to implement for the generative models in this thesis, as fixing the basis functions is equivalent to subtracting their contribution from the observed signal. However further work is required to determine that this inference approach will converge properly, or whether the contribution of shared basis functions to multiple frames should be inferred jointly.

9.3.3 Note Envelopes

We have mostly paid attention to harmonic spectral content of musical notes, which was used solely as a method of multiple pitch detection in Chapter 6. In 5.2.4 we modelled a note as having a constant damping ratio over its length, following from the model of Cemgil et al. [2006], and in 5.2.5 we allow for regular amplitude modulations. In Chapter 8 the excitation vector component of the spectrogram variances was

used to model in a general way the amplitude envelope of a note. However, as discussed in 3.3.1, the note envelope may be divided into attack, sustain, decay and release stages (ASDR), each potentially with a different spectral profile, and other authors such as Orio and Déchelle [2001] have modelled each of these stages as additional states in a hidden Markov model. In Section 8.2 we showed the value in modelling note onsets explicitly as an additional state. Adding additional states to represent decay and release stages is straightforward for both the transcription model in Section 7.5 and the hidden Markov models in Chapter 8.

From a modelling perspective there are still aspects of note envelopes that need further investigation. The onset of a note is very characteristic of the sound of a particular instrument, and hence could be used for instrument identification - an application we have not investigated using the models in this thesis. The onset of a note may also have a percussive component in addition to any harmonic content, which requires an additional component to the harmonic signal model of a musical model, using for example an autoregressive model for the noise process.

A full generative model of a note envelope for different instruments would take into account the relationships between the different stages, including the relative volumes and damping ratios, and the time for each stage.

9.3.4 High Level Score Priors

Score priors represent the highest level of hierarchy in this thesis. They are applied as a prior probability of a pitch being present in a frame, and the probability of a pitch transition from one frame to the next. Having a suitably powerful and realistic prior, which models chords, melodic and bass lines etc. promises to eliminate many transcription errors and increase the accuracy to the level of a human transcriber.

A generative model of a score is ideal, as it can be used as a computer music composition system. Automated music composition using generative models is a popular area of music research. Although many of these models are too intricate for general use as they are designed to emulate a particular genre or style, the basic elements of these models such as chord progressions and the placements of notes around beat positions and divisions of the beat, should be suitable for many applications. One interesting observation is that from a Bayesian perspective, score models with fewer parameters will lead to more consonant and regular sounding music, as notes with multiple shared harmonics often occur together in chords, and regular timings of notes onsets lead to the strong perception of beat and tempo, drawing a parallel with Occam's razor. As music has developed over centuries, harmonic and temporal structures have increased in complexity, so the simplest model may not be appropriate for later genres.

The major challenge with using high level score priors is managing the additional parameters introduced and designing efficient and practical inference algorithms for these models.

Bibliography

S.A. Abdallah and M.D. Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *International Conference on Music Information Retrieval*, 2004.

R. P. Adams, I. Murray, and D. J. C. MacKay. Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

L. Ahlzen and C. Song. *The Sound Blaster Live! Book*. No Starch Press, 2003.

C. Andrieu and A. Doucet. Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, 47:2667–2676, 1999.

I. Arroabarren, M. Zivanovic, X. Rodet, and A. Carlosena. Instantaneous frequency and amplitude of vibrato in singing voice. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003.

M. Barthet, P. Guillemain, R. Kronland-Martinet, and S. Ystad. On the relative influence of even and odd harmonics in clarinet timbre. In *Proc. Int. Comp. Music Conf (ICMC 2005), Barcelona, Spain*, pages 351–354, 2005.

L. Benaroya, R. Gribonval, and F. Bimbot. Non negative sparse representation for Wiener based source separation with a single sensor. 6:613–616, 2003.

N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech and Language Processing*, 18:538–549, 2009a.

N. Bertin, C. Fevotte, and R. Badeau. A tempering approach for Itakura-Saito non-negative matrix factorization. With application to music transcription. In *International Conference on Acoustics, Speech and Signal Processing*, 2009b.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

G. L. Bretthorst. *Bayesian Spectrum Analysis and Parameter Estimation*. Springer-Verlag, 1989.

J. C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89:425–434, 1991.

J. C. Brown and K. V. Vaughn. Pitch center of stringed instrument vibrato tones. *Journal of the Acoustical Society of America*, 100(3):1728–1735, 1996.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

C. Cannam, C. Landone, M. Sandler, and J.P. Bello. The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. In *ISMIR 2006 7th International Conference on Music Information Retrieval Proceedings*, 2006.

P. Cano, A. Loscos, and J. Bonada. Score-performance matching using HMMs. In *International Computer Music Conference*, 1999.

A. Cemgil and O. Dikmen. Conjugate gamma Markov random fields for modelling nonstationary sources. *Independent Component Analysis and Signal Separation*, pages 697–705, 2007.

A. T. Cemgil. Bayesian inference in non-negative matrix factorisation models. Technical report, University of Cambridge, 2008.

A. T. Cemgil and S. J. Godsill. Probabilistic phase vocoder and its application to interpolation of missing values in audio signals. In *13th European Signal Processing Conference*, 2005.

A. T. Cemgil and B. Kappen. Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81, 2003.

A. T. Cemgil, H. J. Kappen, and D. Barber. A generative model for music transcription. *IEEE Transactions on Audio, Speech and Language Processing*, 14:679–694, 2006.

A. T. Cemgil, C. Févotte, and S. J. Godsill. Variational and stochastic inference for Bayesian source separation. *Digital Signal Processing*, 17:891–913, 2007.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90: 75–108, 1995.

S. Chib and I. Jeliazkov. Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, 96:270–281, 2001.

L. Cohen, P. Loughlin, and D. Vakman. On an ambiguity in the definition of the amplitude and phase of a signal. *Signal Processing*, 79:301–307, 1999.

A. Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

A. Cont, D. Schwarz, N. Schnell, and C. Raphael. Evaluation of real-time audio-to-score alignment. In *International Conference on Music Information Retrieval.*, 2007.

Nicholas Cook. *Reactions to the Record: Perspectives on Historical Performance*, chapter Objective expression: phrase arching in recordings of Chopin's Mazurkas.

D. R. Cox and V. Isham. *Point processes*. Chapman & Hall, 1980.

M. J. Crowder, A. C. Kimber, R. L. Smith, and T. J. Sweeting. *Statistical analysis of reliability data*. Chapman & Hall, 1991.

L. Daudet and M. Sandler. MDCT analysis of sinusoids: exact results and applications to coding artifacts reduction. *IEEE Transactions on Speech and Audio Processing*, 12:302–312, 2004.

M. Davy and S. J. Godsill. Bayesian harmonic models for musical signal analysis. In *Bayesian Statistics*. Oxford University Press, 2003.

M. Davy, S. J. Godsill, and J. Idier. Bayesian analysis of polyphonic western tonal music. *Journal of the Acoustical Society of America*, 119:2498–2517, 2006.

N. Degara, A. Pena, M. E. P. Davies, and M. D. Plumbley. Note onset detection using rhythmic structure. In *International Conference on Acoustics, Speech and Signal Processing*, 2010.

J. Devaney, M. I. Mandel, and D. P. W. Ellis. Improving MIDI-audio alignment with acoustic features. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009.

I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. *Advances in neural information processing systems*, 18:283, 2006. ISSN 1049-5258.

O. Dikmen and A. T. Cemgil. Gamma Markov random fields for audio source modeling. *IEEE Transactions on Audio, Speech and Language Processing*, 18:589–601, 2010.

B. N. Dimitrov, VV. Rykov, and Z. L. Krougly. Periodic Poisson processes and almost-lack-of-memory distributions. *Automation and Remote Control*, 65:1597–1610, 2004.

S. Dixon and G. Widmer. Match: A music alignment tool chest. In *Proceedings of the International Conference of Music Information Retrieval*, 2005.

P.M. Djuric. Simultaneous detection and frequency estimation of sinusoidal signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993.

P.M. Djuric. A model selection rule for sinusoids in white Gaussian noise. *IEEE Transactions on Signal Processing*, 44:1744–1751, 1996.

Charles Dodge and Thomas A. Jerse. *Computer Music*. Schirmer Books, 1997.

A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

J. S. Downie. The Music Information Retrieval Evaluation Exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29:247–255, 2008.

D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. *Advances in neural information processing systems*, 20:385–392, 2007.

C. Févotte and A.T. Cemgil. Nonnegative matrix factorizations as probabilistic inference in composite models. In *Proc. 17th European Signal Processing Conf.(EUSIPCO), Glasgow, Scotland*, 2009.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis. *Neural Computation*, 21(3):793–830, March 2009.

J. L. Flanagan, D. I. S. Meinhart, . M. Golden, and M. M. Sondhi. Phase vocoder. *The Journal of the Acoustical Society of America*, 38(5):939–940, 1965.

H. Fletcher and W. A. Munson. Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, 5:82–108, 1933.

N. H. Fletcher and T. D. Rossing. *The physics of musical instruments*. Springer, 1998.

C. Févotte. *Itakura-Saito nonnegative factorizations of the power spectrogram for music signal decomposition*, chapter 11. IGI Global Press, 2010.

D. Gabor. Theory of communication. *IEE Journal on Communications Engineering*, 93:429–457, 1946.

S. A. Gelfand. *Hearing- An Introduction to Psychological and Physiological Acoustics*. Informa HealthCare, 2004.

A. Gelman. *Bayesian data analysis*. CRC press, 2004.

J. M. Geringer and M. L. Allen. An analysis of vibrato among high school and university violin and cello students. *Journal of Research in Music Education*, 52:167–179, 2004.

C. J. Geyer. Reweighting Monte Carlo mixtures. *Journal of Americal Statistical Association*, 1991.

Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. *Advances in Neural Information Processing Systems*, pages 507–513, 2001.

W. R. Gilks and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

S. Godsill. The shifted inverse-gamma model for noise-floor estimation in archived audio recordings. *Signal Processing*, 2009.

S. Godsill and M. Davy. Bayesian harmonic models for musical pitch estimation and analysis. In *IEEE International Conference on Acoustics Speech and Signal Processing*, volume 2, 2002.

S. J. Godsill and M. Davy. Bayesian computational models for inharmonicity in musical instruments. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

S. J. Godsill, A. T. Cemgil, C. Fevotte, and P. J. Wolfe. Bayesian computational methods for sparse audio and music processing. In *15th European Signal Processing Conference*, 2007.

S.J. Godsill. Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes. *International Statistical Review/Revue Internationale de Statistique*, 65(1):1–21, 1997.

M. Goto. Development of the RWC music database. In *Proceedings of the 18th International Congress on Acoustics*, volume 1, pages 553–556, 2004.

M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Music genre database and musical instrument sound database. In *International Symposium on Music Information Retrieval*, pages 229–230, 2003.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711, 1995.

D. D. Greenwood. Critical bandwidth and the frequency coordinates of the basilar membrane. *The Journal of the Acoustical Society of America*, 33:1344, 1961.

D. Guillamet and J. Vitrià. Non-negative matrix factorization for face recognition. *Topics in Artificial Intelligence*, pages 336–344, 2002.

W. M. Hartmann. Pitch, periodicity, and auditory organization. *The Journal of the Acoustical Society of America*, 100:3491, 1996.

N. Hu, R. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

F. Itakura and S. Saito. Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the 6th International Congress on Acoustics*, 1968.

J. S. R. Jang, H. R. Lee, and C. H. Yeh. Query by tapping: A new paradigm for content-based music retrieval from acoustic input. In *IEEE Pacific Rim Conference on Multimedia*, 2001.

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 453–461, 1946.

M. Karjalainen and U. K. Laine. A model for real-time sound synthesis of guitar on a floating-point signal processor. In *International Conference on Acoustics, Speech, and Signal Processing*, 1991.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430), 1995.

A. Klapuri. *Signal Processing Methods for Music Transcription*, chapter Auditory-Model Based Methods for Multiple F0 Estimation, pages 229–265. Springer, 2006.

A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16:255–266, 2008.

A. P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, 2003.

A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14:342–355, 2006.

A.P. Klapuri. Automatic music transcription as we know it today. *Journal of New Music Research*, 33(3): 269–282, 2004.

H. Lantéri, C. Theys, C. Richard, and C. Févotte. Split gradient method for nonnegative matrix factorization. In *European Signal Processing Conference*, Aalborg, Denmark, Aug. 2010.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing*, 2000.

C.J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18(6):1589–1596, 2007a. ISSN 1045-9227.

C.J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10): 2756–2779, 2007b. ISSN 0899-7667.

J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2003.

R. B. MacLeod. Influences of dynamic level and pitch register on the vibrato rates and widths of violin and viola players. *Journal of Research in Music Education*, 56:43–54, 2008.

R. C. Maher and J. W. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 4:2254–2263, 1995.

S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.

M. Marolt. A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Trans. Multimedia*, 6(3):439–449, Jun. 2004.

R. Martin. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing*, 9, 2001.

R. Meddis and L. O' Mard. A unitary model of pitch perception. *The Journal of the Acoustical Society of America*, 102:1811, 1997.

W. Q. Meeker and L. A. Escobar. *Statistical methods for reliability data*. Wiley, 1998.

J. A. Moorer. On the transcription of musical sound by computer. *Journal of Computer Music*, pages 32–38, 1977.

F. D. Neeser and J. L. Massey. Proper complex random processes with applications to information theory. *IEEE Transactions on Information Theory*, 39(4):1293–1302, 1993.

B. Niedermayer. Towards audio to score alignment in the symbolic domain. In *Sound and Music Computing Conference*, 2009.

N. Orio and F. Déchelle. Score following using spectral analysis and hidden Markov models. In *Proceedings of the ICMC*, pages 151–154, 2001.

N. Orio and D. Schwarz. Alignment of monophonic and polyphonic music to a score. In *International Computer Music Conference*, 2001.

A. Ozerov, C. Févotte, and M. Charbit. Factorial scaled hidden Markov model for polyphonic audio representation and source separation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, NY, USA, Oct. 2009.

P. Paatero. Least squares formulation of robust nonnegative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37:22–35, 1997.

H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 53–60, 2007.

R. D Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand. Complex sounds and auditory images. *Auditory physiology and perception*, 83:429–446, 1992.

J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *European Signal Processing Conference*, 2006.

P. H. Peeling, A. T. Cemgil, and S. J. Godsill. A probabilistic framework for matching music representations. In *International Conference on Music Information Retrieval*, 2007a.

P. H. Peeling, C. F. Li, and S. J. Godsill. Poisson point process modeling for polyphonic music transcription. *Journal of the Acoustical Society of America*, 121:EL168–EL175, 2007b.

P.H. Peeling, A.T. Cemgil, and S.J. Godsill. Generative spectrogram factorization models for polyphonic piano transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18:519–527, 2010. ISSN 1558-7916.

G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 115–120. Citeseer, 2006.

A. Pertusa and J. M. Inesta. Multiple fundamental frequency estimation using Gaussian smoothness. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

B. Picinbono and P. Bondon. Second-order statistics of complex signals. *IEEE Transactions on Signal Processing*, 45(2):411–420, 1997.

C. J. Plack, A. J. Oxenham, and R. R. Fay, editors. *Pitch: Neural Coding and Perception*. Springer, 2005.

M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler. Automatic music transcription and audio source separation. *Cybernetics and Systems*, 33(6):603–627, 2002.

G. E. Poliner and D. P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007.

E. Prame. Measurements of the vibrato rate of ten singers. *Journal of the Acoustical Society of America*, 96 (4):1979–1984, 1994.

J. P. Princen, A. W. Johnson, and A. B. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1987.

M. Puckette, T. Apel, and D. Zicarelli. Real-time audio analysis tools for Pd and MSP. In *International Computer Music Conference*, 1998.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

L. R. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice-Hall, 1993.

C. Raphael. Automated rhythm transcription. In *International Symposium on Music Information Retrieval*, 2001.

C. Raphael. A hybrid graphical model for aligning polyphonic audio with musical scores. In *International Conference on Musical Information Retrieval*, 2004.

C. Raphael. Aligning music audio with symbolic scores using a hybrid graphical model. *Machine Learning*, 65:389–409, 2006.

C. Raphael. A classifier-based approach to score-guided source separation of musical audio. *Computer Music Journal*, 32(1):51–59, 2008.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with unknown number of components. *Journal of the Royal Statistical Society: Series B*, 4:731–792, 1997.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2004.

A. Robertson and M. D. Plumbley. Post-processing fiddle: A real-time multi-pitch tracking technique using harmonic partial subtraction for use within live performance systems. In *International Computer Music Conference*, 2009.

M. P. Ryynänen and A. P. Klapuri. Modelling of note events for singing transcription. In *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*. Citeseer, 2004.

M. P. Ryynänen and A. P. Klapuri. Polyphonic music transcription using note event modeling. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

E. D. Scheirer. *Readings in Computational Auditory Scene Analysis*, chapter Using musical knowledge to extract expressive performance information from audio recordings, pages 361–380. L. Erlbaum Associates Inc., 1998.

E. D. Scheirer. *Music-listening systems*. PhD thesis, Massachusetts Institute of Technology, 2000.

M. N. Schmidt and H. Laurberg. Nonnegative matrix factorization with Gaussian process priors. *Computational intelligence and neuroscience*, 2008.

D. Schwarz, A. Cont, and N. Schnell. From Boulez to ballads: Training IRCAM's score follower. In *Proceedings of International Computer Music Conference*, 2005.

C. Seashore. *Objective analysis of musical performance*. McGraw-Hill, 1936.

X. Serra. Musical sound modeling with sinusoids plus noise. *Musical signal processing*, pages 497–510, 1997.

C. E. Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.

P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

A. Stark and M. D. Plumbley. Tracking a performance without a score. In *International Conference on Acoustics, Speech and Signal Processing*, 2010.

S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8:185, 1937.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 611–622, 1999.

P. M. Todd and D. G. Loy. *Music and connectionism*. The MIT Press, 1991.

T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716, 2000.

R. J. Turetsky and D. P. W. Ellis. Ground-truth transcriptions of real music from force-aligned MIDI syntheses. In *International Conference on Music Information Retrieval*, 2003.

R. Typke and A. Walczak-Typke. A tunneling-vantage indexing method for non-metrics. In *International Conference on Music Information Retrieval*, 2008.

B. L. Vercoe, W. G. Gardner, and E. D. Scheirer. Structured audio: Creation, transmission, and rendering of parametric sound representations. *Proceedings of the IEEE*, 86(5):922–940, 1998.

E. Vincent, N. Bertin, and R. Badeau. Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *IEEE Internation Conference on Acoustics, Speech and Signal Processing*, 2008.

E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):528–537, 2010. ISSN 1558-7916.

T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007.

T. Virtanen, A. T. Cemgil, and S. J. Godsill. Bayesian extensions to non-negative matrix factorisation for audio signal modelling. In *International Conference on Acoustics, Speech and Signal Processing*, 2008.

G. H. Wakefield. Mathematical representation of joint time-chroma distributions. In *Proceedings of SPIE*, volume 3807, page 637, 1999.

P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 119–122. Citeseer, 1999.

B. Wang and M.D. Plumbley. Musical audio stream separation by non-negative matrix factorization. In *DMRN Summer Conference*, 2005.

N. P. Whiteley, A. T. Cemgil, and S. J. Godsill. Bayesian modelling of temporal structure in musical audio. In *International Conference on Music Information Retrieval*, 2006.

P. J. Wolfe, M. Dorfler, and S. J. Godsill. Bayesian modelling of time-frequency coefficients for audio signal enhancement. *Advances in Neural Information Processing Systems*, 2003.

P. J. Wolfe, S. J. Godsill, and W. J. Ng. Bayesian variable selection and regularization for time-frequency surface estimation. *Journal of the Royal Statistical Society Series B*, 66:575–589, 2004.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 267–273. ACM, 2003. ISBN 1581136463.

C. Yeh, A. Röbel, and X. Rodet. Multiple fundamental frequency estimation of polyphonic music signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.

A. Zellner. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, 6:233–243, 1986.

F. Zhang, G. Bi, and Y. Q. Chen. Harmonic transform. *IEE Proceedings-Vision, Image and Signal Processing*, 151(4):257–263, 2004.

E. Zwicker. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *Acoustical Society of America Journal*, 33:248, 1961.

Appendix A

Probability Distributions

A.1 Normal Distribution

The normal distribution $\mathcal{N}\left(x;\mu,\sigma^2\right)$, for real valued x with mean μ and variance σ^2 has probability distribution

$$p\left(x\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\frac{\left(x-\mu\right)^2}{\sigma^2}\right)$$

μ is a location parameter for the normal distribution, and σ^2 is a scale parameter. Sufficient statistics for a set of normally distributed observations x_i are given by $\sum x_i$ and $\sum x_i^2$. Maximum likelihood estimates of the parameters are given by $\hat{\mu}=\sum x_i$ and $\hat{\sigma^2}=\sum x_i^2-\hat{\mu}^2$. The entropy of the normal distribution is $\frac{1}{2}\ln\left(2\pi e\sigma^2\right)$.

The normal distribution is the conjugate prior distribution for the mean parameter of a normal distribution. If $\mu\sim\mathcal{N}\left(\mu_0,\sigma_0^2\right)$ and the variance σ^2 is known, then the posterior distribution of μ given n observations x_i is a normal distribution with mean

$$\left(\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}\right)^{-1}\left(\frac{\mu_0}{\sigma_0^2}+\frac{\sum x_i}{\sigma^2}\right)$$

and variance

$$\left(\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}\right)^{-1}$$

If x is a complex number where the real and imaginary components are independently normally distributed with zero mean and variance σ^2, then

$$p\left(x\right)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2}\frac{\left|x\right|^2}{\sigma^2}\right)$$

A.2 Gamma Distribution

The Gamma distribution $\mathcal{G}(r;\alpha,\beta)$ for $r>0$ where $\alpha>0$ is the shape parameter and $\beta>0$ is a scale parameter has probability distribution

$$p(r)=r^{\alpha-1}\frac{\exp(-r/\beta)}{\beta^{\alpha}\Gamma(\alpha)}$$

Sufficient statistics for a set of Gamma distributed observations r_i are $\sum r_i$ and $\sum_i \log r_i$. The parameters of the Gamma distribution are related to the sufficient statistics by

$$\begin{aligned}\frac{1}{N}\sum_i \log r_i &= \psi(a)-\log\beta\\ \frac{1}{N}\sum_i r_i &= ab\end{aligned}$$

The entropy of the gamma distribution is

$$\alpha+\ln\beta+\ln\Gamma(\alpha)+(1-\alpha)\psi(\alpha)$$

The gamma distribution is the conjugate prior for the rate parameter of the Poisson distribution. If $\lambda\sim\mathcal{G}(\alpha,\beta)$ and we have n observations $x_i\sim\mathcal{P}o(\lambda)$ then the posterior distribution of λ is

$$p(\lambda|x_1,\ldots,x_n,\alpha,\beta)=\mathcal{G}\left(\alpha+\sum_i x_i,\beta+n\right)$$

A.3 Inverse-Gamma Distribution

The inverse Gamma distribution $\mathcal{IG}(r;\alpha,\beta)$ for $r>0$ where $\alpha>0$ is the shape parameter and $\beta>0$ is a scale parameter has probability distribution

$$p(r)=(1/r)^{\alpha-1}\,\frac{\beta^{\alpha}\exp(-\beta/r)}{\Gamma(\alpha)}$$

The inverse Gamma distribution is the distribution of the random variable $1/r$ when $r\sim\mathcal{G}(r;\alpha,1/\beta)$. Sufficient statistics for a set of inverse Gamma distributed observations r_i are $\sum r_i^{-1}$ and $\sum_i \log r_i$. The parameters of the inverse Gamma distribution are related to the sufficient statistics by

$$\begin{aligned}\frac{1}{N}\sum_i \log r_i &= -\psi(a)+\log\beta\\ \frac{1}{N}\sum_i r_i &= a/b\end{aligned}$$

The entropy of the inverse Gamma distribution is

$$\alpha+\ln\beta+\ln\Gamma(\alpha)-(1-\alpha)\psi(\alpha)$$

The inverse Gamma distribution is the conjugate prior distribution for the variance parameter of a normal distribution. If $\sigma^2 \sim \mathcal{IG}(\alpha, \beta)$ and the mean μ is known, then the posterior distribution of σ^2 given n observations x_i is

$$p\left(\sigma^2 | x_1, \ldots, x_n, \mu, \alpha, \beta\right) = \mathcal{IG}\left(\alpha + \frac{n}{2}, \beta + \frac{\sum_i (x_i - \mu)^2}{2}\right)$$

A.4 Beta Distribution

The Beta distribution $\mathcal{B}eta(x; \alpha, \beta)$ where $0 \leq x \leq 1$ and $\alpha > 0, \beta > 0$ has probability distribution

$$p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

The Beta distribution is the conjugate prior to the probability of success ρ of a set of n Bernoulli trials r_i where $r_i \in \{0, 1\}$. The posterior distribution of ρ is

$$p(\rho | r_1, \ldots, r_n, \alpha, \beta) = \mathcal{B}eta\left(\alpha + \sum_i r_i, \beta + n - \sum_i r_i\right)$$

Appendix B

Derivation of Results

B.1 Mode of Posterior Distribution of Signal-to-noise Parameter

Take the natural log of (5.22)

$$\log p\left(y|\mathbf{D},\xi\right)+\log p\left(\xi\right)=-\frac{N+\alpha_n}{2}\log\left(y^\top\mathbf{P}y+\beta_n\right)-\left(\alpha_\xi+1\right)\log\left(\xi\right)-\frac{\beta_\delta}{\xi}$$

differentiate the expression with respect to δ^2 and set equal to zero:

$$\begin{aligned}
-\frac{N+\alpha_n}{2}\frac{-\frac{1}{\xi+1}y^\top\mathbf{D}\mathbf{D}^\dagger y}{y^\top y-\frac{\xi}{\xi+1}y^\top\mathbf{D}\mathbf{D}^\dagger y}-\frac{\alpha_\xi+1}{\xi}+\frac{\beta_\xi}{\xi^2} &= 0\\
\frac{N+\alpha_n}{2}\frac{y^\top\mathbf{D}\mathbf{D}^\dagger y}{y^\top y-\xi y^\top\mathbf{D}\mathbf{D}^\dagger y}-\frac{\alpha_\xi+1}{\xi}+\frac{\beta_\xi}{\xi^2} &= 0\\
\frac{N+\alpha_n}{2}\frac{\xi^2 y^\top\mathbf{D}\mathbf{D}^\dagger y}{y^\top y-\xi y^\top\mathbf{D}\mathbf{D}^\dagger y}-\xi\left(\alpha_\xi+1\right)+\beta_\xi &= 0
\end{aligned}$$

After some rearranging

$$\begin{aligned}
\frac{N+\alpha_n}{2}\left(\delta^2\right)^2 y^\top\mathbf{D}\mathbf{D}^\dagger y+\left(\beta_\xi-\delta^2\left(\alpha_\xi+1\right)\right)\left(y^\top y-\delta^2 y^\top\mathbf{D}\mathbf{D}^\dagger y\right) &= 0\\
\xi^2\left(\frac{N+\alpha_n}{2}+\left(\alpha_\xi+1\right)\right)\left(y^\top\mathbf{D}\mathbf{D}^\dagger y\right)-\xi\left(\left(\alpha_\xi+1\right)y^\top y+\beta_\xi y^\top\mathbf{D}\mathbf{D}^\dagger y\right)+\beta_\xi y^\top y &= 0 \qquad \text{(B.1)}
\end{aligned}$$

B.2 Posterior over Latent Sources in Gaussian Variance Matrix Factorization Model

J is the identity matrix with dimensions $I \times I$

$$
\begin{aligned}
&-\frac{D}{2}\mathrm{Tr}A^{-1}ss^H + \frac{D}{2}\mathrm{Tr}\frac{1^\top 1 ss^H}{1A1^\top} + \cdots \\
&= -\frac{D}{2}\mathrm{Tr}A^{-1}\left(J - A\frac{1^\top 1}{1A1^\top}\right)ss^H + \cdots \\
&= -\frac{D}{2}\mathrm{Tr}\frac{A^{-1}}{1A1^\top}(1A1^\top J - A1^\top 1)ss^H + \cdots \\
&= -\frac{D}{2}\mathrm{Tr}\frac{1}{1A1^\top}(1A1^\top J - A1^\top 1)^\top (1A1^\top J - A1^\top 1)^{-1}A^{-1}(1A1^\top J - A1^\top 1)ss^H + \ldots \\
&= -\frac{D}{2}\mathrm{Tr}\left(s - \frac{A1^\top 1s}{1A1^\top}\right)^H (1A1^\top J - A1^\top 1)^{-1}\frac{A^{-1}}{1A1^\top}\left(s - \frac{A1^\top 1s}{1A1^\top}\right) + \ldots \\
&= -\frac{D}{2}\mathrm{Tr}\left(s - \frac{A1^\top 1s}{1A1^\top}\right)^H \left(A - \frac{A1^\top 1A}{1A1^\top}\right)^{-1}\left(s - \frac{A1^\top 1s}{1A1^\top}\right) + \ldots
\end{aligned}
$$