

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/59219>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

Bayesian Music Transcription

Een wetenschappelijke proeve op het gebied van de
Natuurwetenschappen, Wiskunde en Informatica

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de Rector Magnificus prof. dr. C.W.P.M. Blom,
volgens besluit van het College van Decanen
in het openbaar te verdedigen
op dinsdag 14 September 2004
des namiddags om 1:30 uur precies

door

Ali Taylan Cemgil

geboren op 28 Januari 1970 te Ankara, Turkije

Promotor : prof. dr. C.C.A.M. Gielen

Co-promotor : dr. H. J. Kappen

Manuscript commissie : dr. S.J. Godsill, University of Cambridge
prof. dr. F.C.A. Groen, University of Amsterdam
prof. dr. M. Leman, University of Gent

©2004 Ali Taylan Cemgil

ISBN 90-9018454-6

The research described in this thesis is fully supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs.

Contents

1	Introduction	1
1.1	Rhythm Quantization and Tempo Tracking	3
1.1.1	Related Work	6
1.2	Polyphonic Pitch Tracking	6
1.2.1	Related Work	7
1.3	Probabilistic Modelling and Music Transcription	9
1.3.1	Bayesian Inference	9
1.4	A toy example	11
1.5	Outline of the thesis	13
1.6	Future Directions and Conclusions	13
2	Rhythm Quantization	15
2.1	Introduction	15
2.2	Rhythm Quantization Problem	16
2.2.1	Definitions	16
2.2.2	Performance Model	19
2.2.3	Bayes Theorem	19
2.2.4	Example 1: Scalar Quantizer (Grid Quantizer)	20
2.2.5	Example 2: Vector Quantizer	21
2.3	Verification of the Model	24
2.3.1	Perception Task	25
2.3.2	Production Task	26
2.3.3	Estimation of model parameters	27
2.3.4	Results	28
2.4	Discussion and Conclusion	30
2.A	Estimation of the posterior from subject responses	31
3	Tempo Tracking	33
3.1	Introduction	33
3.2	Dynamical Systems and the Kalman Filter	34
3.2.1	Extensions	36
3.3	Tempogram Representation	36
3.4	Model Training	39
3.4.1	Estimation of τ_j from performance data	40
3.4.2	Estimation of state transition parameters	40
3.4.3	Estimation of switch parameters	40
3.4.4	Estimation of Tempogram parameters	41
3.5	Evaluation	41
3.5.1	Data	41

3.5.2	Kalman Filter Training results	42
3.5.3	Tempogram Training Results	43
3.5.4	Initialization	43
3.5.5	Evaluation of tempo tracking performance	43
3.5.6	Results	44
3.6	Discussion and Conclusions	45
4	Integrating Tempo Tracking and Quantization	47
4.1	Introduction	47
4.2	Model	49
4.2.1	Score prior	50
4.2.2	Tempo prior	51
4.2.3	Extensions	52
4.2.4	Problem Definition	53
4.3	Monte Carlo Simulation	55
4.3.1	Simulated Annealing and Iterative Improvement	56
4.3.2	The Switching State Space Model and MAP Estimation	56
4.3.3	Sequential Monte Carlo	58
4.3.4	SMC for the Switching State Space Model	60
4.3.5	SMC and estimation of the MAP trajectory	61
4.4	Simulations	62
4.4.1	Artificial data: Clave pattern	62
4.4.2	Real Data: Beatles	63
4.5	Discussion	67
4.A	A generic prior model for score positions	72
4.B	Derivation of two pass Kalman filtering Equations	72
4.B.1	The Kalman Filter Recursions	73
4.C	Rao-Blackwellized SMC for the Switching State space Model	75
5	Piano-Roll Inference	77
5.1	Introduction	77
5.1.1	Music Transcription	77
5.1.2	Approach	78
5.2	Polyphonic Model	79
5.2.1	Modelling a single note	80
5.2.2	From Piano-Roll to Microphone	81
5.2.3	Inference	83
5.3	Monophonic Model	84
5.4	Polyphonic Inference	87
5.4.1	Vertical Problem: Chord identification	88
5.4.2	Piano-Roll inference Problem: Joint Chord and Melody identification	91
5.5	Learning	91
5.6	Discussion	94
5.6.1	Future work	97
5.A	Derivation of message propagation algorithms	98
5.A.1	Computation of the evidence $p(y_{1:T})$	98
5.6.2	Computation of MAP configuration $r_{1:T}^*$	100
5.A.3	Inference for monophonic pitch tracking	101
5.A.4	Monophonic pitch tracking with varying fundamental frequency	102
5.B	Computational Simplifications	103

<i>CONTENTS</i>	iii
5.B.1 Pruning	103
5.B.2 Kalman filtering in a reduced dimension	103
Publications	105
Bibliography	107
Samenvatting	115
Dankwoord	117
Curriculum Vitae	119

Chapter 1

Introduction

Music transcription refers to extraction of a human readable and interpretable description from a recording of a music performance. The interest into this problem is mainly motivated by the desire to implement a program to infer automatically a musical notation (such as the traditional western music notation) that lists the pitch levels of notes and corresponding timestamps in a given performance.

Besides being an interesting problem of its own, automated extraction of a score (or a score-like description) is potentially very useful in a broad spectrum of applications such as interactive music performance systems, music information retrieval and musicological analysis of musical performances. However, in its most unconstrained form, i.e., when operating on an arbitrary acoustical input, music transcription stays yet as a very hard problem and is arguably “AI-complete”, i.e. requires simulation of a human-level intelligence. Nevertheless, we believe that an eventual practical engineering solution is possible by an interplay of scientific knowledge from cognitive science, musicology, musical acoustics and computational techniques from artificial intelligence, machine learning and digital signal processing. In this context, the aim of this thesis is to integrate this vast amount of prior knowledge in a consistent and transparent computational framework and to demonstrate the feasibility of such an approach in moving us closer to a practical solution to music transcription.

In a statistical sense, music transcription is an inference problem where, given a signal, we want to find a score that is consistent with the encoded music. In this context, a score can be contemplated as a collection of “musical objects” (e.g., note events) that are rendered by a performer to generate the observed signal. The term “musical object” comes directly from an analogy to visual scene analysis where a scene is “explained” by a list of objects along with a description of their intrinsic properties such as shape, color or relative position. We view music transcription from the same perspective, where we want to “explain” individual samples of a music signal in terms of a collection of musical objects where each object has a set of intrinsic properties such as pitch, tempo, loudness, duration or score position. It is in this respect that a score is a high level description of music.

Musical signals have a very rich temporal structure, and it is natural to think of them as being organized in a hierarchical way. On the highest level of this organization, which we may call as the cognitive (symbolic) level, we have a score of the piece, as, for instance, intended by a composer¹. The performers add their interpretation to music and render the score into a collection of “control signals”. Further down on the physical level, the control signals trigger various musical instruments that synthesize the actual sound signal. We illustrate these generative processes using a hierarchical graphical model (See Figure 1.1), where the arcs represent generative links.

¹In reality the music may be improvised and there may be actually not a written score. However, for doing transcription we have to assume the existence a score as our starting point.

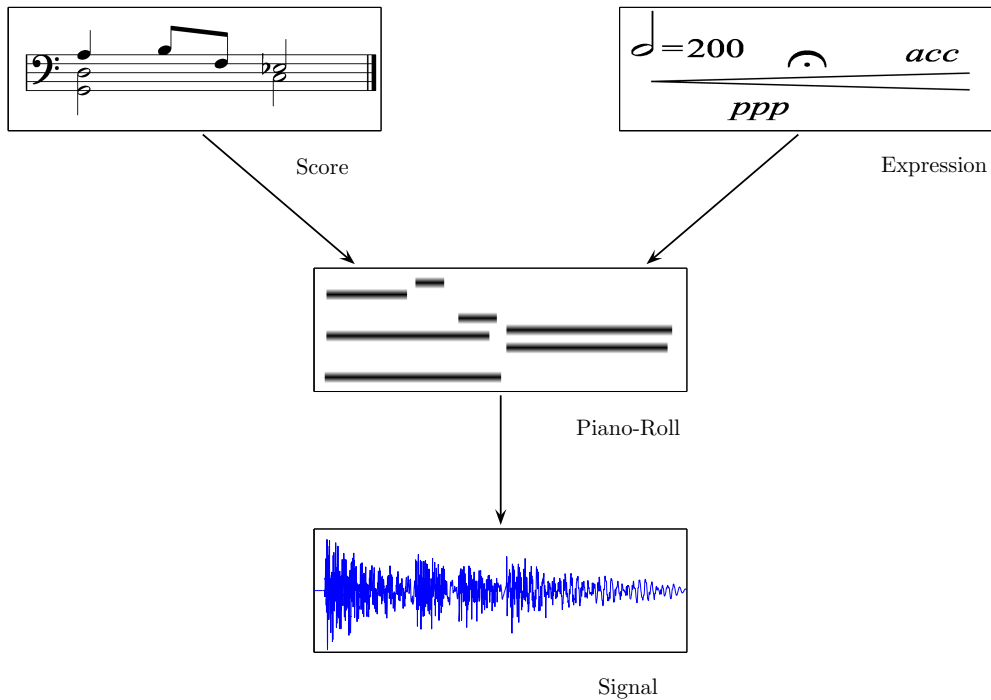


Figure 1.1: A hierarchical generative model for music signals. In this model, an unknown score is rendered by a performer into a piano-roll. The performer introduces expressive timing deviations and tempo fluctuations. The piano-roll is rendered into audio by a synthesis model. The piano roll can be viewed as a symbolic representation, analogous to a sequence of MIDI events. Given the observations, transcription can be viewed as inference of the score by “inverting” the model. Somewhat simplified, the transcription methods described in this thesis can be viewed as inference techniques as applied to subgraphs of this graphical model. Rhythm quantization (Chapter 2) is inference of the score given onsets from a piano-roll (i.e. a list of onset times) and tempo. Tempo tracking, as described in Chapter 3 corresponds to inference of the expressive deviations introduced by the performer, given onsets and a score. Joint quantization and tempo tracking (Chapter 4) infers both the tempo and score simultaneously, given only onsets. Polyphonic pitch tracking (Chapter 5) is inference of a piano-roll given the audio signal.

This architecture is of course anything but new, and in fact underlies any music generating computer program such as a sequencer. The main difference of our model from a conventional sequencer is that the links are probabilistic, instead of deterministic. We use the sequencer analogy in describing a realistic generative process for a large class of music signals.

In describing music, we are usually interested in a symbolic representation and not so much in the “details” of the actual waveform. To abstract away from the signal details, we define an intermediate layer, that represent the control signals. This layer, that we call a “piano-roll”, forms the interface between a symbolic process and the actual signal process. Roughly, the symbolic process describes how a piece is composed and performed. Conditioned on the piano-roll, the signal process describes how the actual waveform is synthesized. Conceptually, the transcription task is then to “invert” this generative model and recover back the original score.

In the next section, we will describe three subproblems of music transcription in this framework. First we introduce models for *Rhythm Quantization* and *Tempo Tracking*, where we assume that exact timing information of notes is available, for example as a stream of MIDI² events from a digital keyboard. In the second part, we focus on *polyphonic pitch tracking*, where we estimate note events from acoustical input.

1.1 Rhythm Quantization and Tempo Tracking

In conventional music notation, the onset time of each note is implicitly represented by the cumulative sum of durations of previous notes. Durations are encoded by simple rational numbers (e.g., quarter note, eighth note), consequently all events in music are placed on a discrete grid. So the basic task in MIDI transcription is to associate onset times with discrete grid locations, i.e., quantization.

However, unless the music is performed with mechanical precision, identification of the correct association becomes difficult. This is due to the fact that musicians introduce intentional (and unintentional) deviations from a mechanical prescription. For example timing of events can be deliberately delayed or pushed. Moreover, the tempo can fluctuate by slowing down or accelerating. In fact, such deviations are natural aspects of expressive performance; in the absence of these, music tends to sound rather dull and mechanical. On the other hand, if these deviations are not accounted for during transcription, resulting scores have often very poor quality. Figure 1.2 demonstrates an instance of this.

A computational model for tempo tracking and transcription from a MIDI-like music representation is useful in automatic score typesetting, the musical analog of word processing. Almost all score typesetting applications provide a means of automatic generation of a conventional music notation from MIDI data. Robust and fast quantization and tempo tracking is also an important requirement for interactive performance systems; applications that “listen” to a performer for generating an accompaniment or improvisation in real time (Raphael, 2001b; Thom, 2000).

From a theoretical perspective, simultaneous quantization *and* tempo tracking is a “chicken-and-egg” problem: the quantization depends upon the intended tempo interpretation and the tempo interpretation depends upon the quantization (See Figure 1.3).

Apparently, human listeners can resolve this ambiguity in most cases without much effort. Even persons without any musical training are able to determine the beat and the tempo very rapidly. However, it is still unclear what precisely constitutes tempo and how it relates to the

²Musical Instruments Digital Interface. A standard communication protocol especially designed for digital instruments such as keyboards. Each time a key is pressed, a MIDI keyboard generates a short message containing pitch and key velocity. A computer can tag each received message by a timestamp for real-time processing and/or recording into a file.

Prelude in C major
BWV 846

Allegro ($\text{♩} = 112$) J. S. Bach

The image shows the original musical score for the C major prelude (BWV 846) by J.S. Bach. It consists of two systems of music. The first system is marked 'Allegro' with a tempo of quarter note = 112. The second system is marked 'cresc.'. The score is written for piano and includes performance instructions like 'legato'.

(a) Original Score

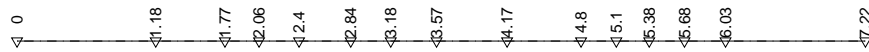
The image shows a transcription of the C major prelude without tempo tracking. The score is written in three systems and is highly complex and difficult to read due to irregular note spacing and timing.

(b) Transcription without tempo tracking

The image shows a transcription of the C major prelude produced by the system. The score is written in three systems and shows a clear and simple rhythmic structure.

(c) Output of our system

Figure 1.2: Excerpts from a performance of the C major prelude (BWV 846 - first book of the well tempered clavier). A pianist is invited to play the original piece in Figure (a) on a digital MIDI piano. He was free in choosing any interpretation. We can transcribe the performance directly using a conventional music typesetting program; however the resulting score becomes rapidly very complex and useless for a human reader (Figure (b)). This is primarily due to the fact that tempo fluctuations and expressive timing deviations are not accounted for. Consequently, the score does not display the simple regular rhythmical structure of the piece. In Figure (c), a transcription is shown that is produced by our system that displays the simple rhythmical structure.



(a) Example: A performed onset sequence

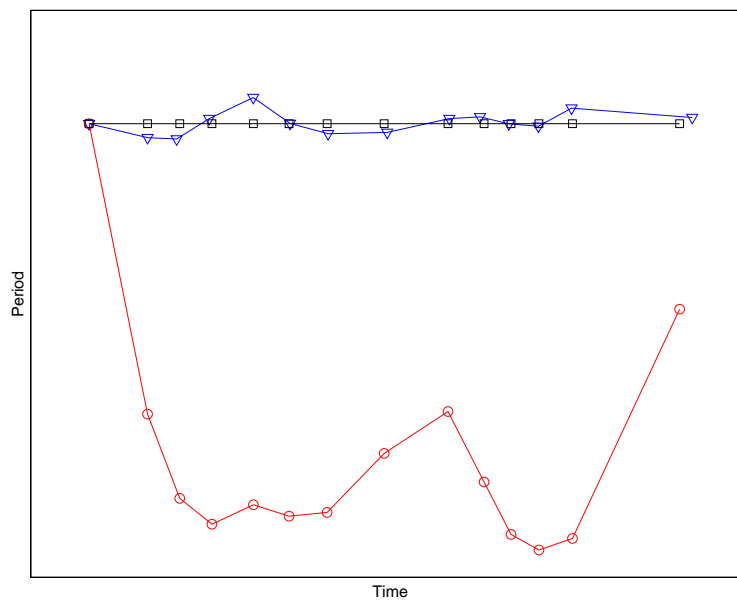


(b) “Too” accurate quantization. Although the resulting notation represents the performance well, it is unacceptably complicated.

(c) “Too” simple notation. This notation is simpler but is a very poor description of the rhythm.



(d) Desired quantization balances accuracy and simplicity.



(e) Corresponding tempo-curves. Curves with square, oval and triangle dots correspond to the notation 1.3(b), 1.3(c) and 1.3(d).

Figure 1.3: The tradeoff between quantization and tempo tracking. Given any sequence of onset times, we can in principle easily find a notation (i.e. a sequence of rational numbers) to describe the timing information arbitrarily well. Consider the performed simple rhythm in 1.3(a) (from Desain & Honing, 1991). A very fine grid quantizer produces a result similar to 1.3(b). Although this is a very accurate representation, the resulting notation is far too complex. Another extreme case is the notation in 1.3(c), that contains notes of equal duration. Although this notation is very “simple”, it is very unlikely that it is the intended score, since this would imply that the performer has introduced very unrealistic tempo changes (See 1.3(e)). Musicians would probably agree that the “smoother” score shown in 1.3(d) is a better representation. This example suggests that a *good score* must be “easy” to read while representing the timing information accurately.

perception of the beat, rhythmical structure, pitch, style of music etc. Tempo is a perceptual construct and cannot directly be measured in a performance.

1.1.1 Related Work

The goal of understanding tempo perception has stimulated a significant body of research on psychological and computational modelling aspects of tempo tracking and beat induction. Early work by (Michon, 1967) describes a systematic study on the modelling of human behaviour in tracking tempo fluctuations in artificially constructed stimuli. (Longuet-Higgins, 1976) proposes a musical parser that produces a metrical interpretation of performed music while tracking tempo changes. Knowledge about meter helps the tempo tracker to quantize a performance.

Large and Jones (1999) describe an empirical study on tempo tracking, interpreting the observed human behaviour in terms of an oscillator model. A peculiar characteristic of this model is that it is insensitive (or becomes so after enough evidence is gathered) to material in between expected beats, suggesting that the perception tempo change is indifferent to events in this interval. (Toiviainen, 1999) discusses some problems regarding phase adaptation.

Another class of tempo tracking models are developed in the context of interactive performance systems and score following. These models make use of prior knowledge in the form of an annotated score (Dannenberg, 1984; Vercoe & Puckette, 1985). More recently, Raphael (2001b) has demonstrated an interactive real-time system that follows a solo player and schedules accompaniment events according to the player's tempo interpretation.

More recently attempts are made to deal directly with the audio signal (Goto & Muraoka, 1998; Scheirer, 1998) without using any prior knowledge. However, these models assume constant tempo (albeit timing fluctuations may be present). Although successful for music with a steady beat (e.g., popular music), they report problems with syncopated data (e.g., reggae or jazz music).

Many tempo tracking models assume an initial tempo (or beat length) to be known to start up the tempo tracking process (e.g., (Longuet-Higgins, 1976; Large & Jones, 1999). There is few research addressing how to arrive at a reasonable first estimate. (Longuet-Higgins & Lee, 1982) propose a model based on score data, (Scheirer, 1998) one for audio data. A complete model should incorporate both aspects.

Tempo tracking is crucial for quantization, since one can not uniquely quantize onsets without having an estimate of tempo and the beat. The converse, that quantization can help in identification of the correct tempo interpretation has already been noted by Desain and Honing (1991). Here, one defines correct tempo as the one that results in a simpler quantization. However, such a schema has never been fully implemented in practice due to computational complexity of obtaining a perceptually plausible quantization. Hence quantization methods proposed in the literature either estimate the tempo using simple heuristics (Longuet-Higgins, 1987; Pressing & Lawrence, 1993; Agon, Assayag, Fineberg, & Rueda, 1994) or assume that the tempo is known or constant (Desain & Honing, 1991; Cambouropoulos, 2000; Hamanaka, Goto, Asoh, & Otsu, 2001).

1.2 Polyphonic Pitch Tracking

To transcribe a music performance from acoustical input, one needs a mechanism to sense and characterize individual events produced by the instrumentalist. One potential solution is to use dedicated hardware and install special sensors on to the instrument body: this solution has restricted flexibility and is applicable only to instruments designed specifically for such a purpose. Discounting the 'hardware' solution, we shall assume that we capture the sound with a single microphone, so that the computer receives no further input other than the pure acoustic information. In this context, polyphonic pitch tracking refers to identification of (possibly simultaneous)

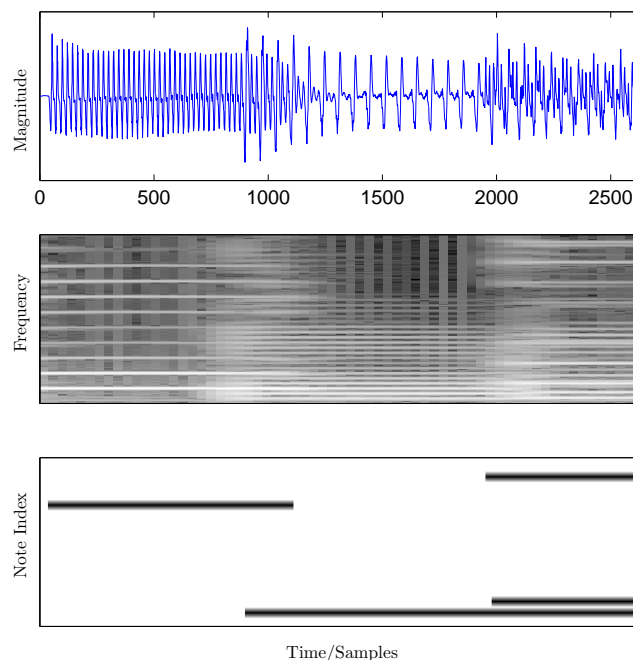


Figure 1.4: Piano Roll inference from polyphonic signals. (Top) A short segment of the polyphonic music signal. (Middle) Spectrogram (Magnitude of the Short time Fourier transform) of the signal. Horizontal and vertical axes correspond to time and frequency, respectively. Grey level denotes the energy in a logarithmic scale. The line spectra (parallel “lines” to time axis equispaced in frequency) are characteristic to many pitched musical signals. The low frequency notes are not well resolved due to short window length. Taking a longer analysis window would increase the frequency resolution but smear out onsets and offsets. When two or more notes are played at the same time, their harmonics overlap both in time and frequency, making correct associations of individual harmonics to note events difficult. (Bottom) A “piano-roll” denoting the note events where the vertical axis corresponds to the note index and the horizontal axis corresponds to time index. Black and white pixels correspond to “sound” and “mute” respectively. The piano-roll can be viewed as a symbolic summary of the underlying signal process.

note events. The main challenge is separation and identification of typically small (but unknown) number of source signals that overlap both in time and frequency (See Figure 1.4).

1.2.1 Related Work

Polyphonic pitch identification has attracted quite an amount of research effort in the past; see (Plumbley, Abdallah, Bello, Davies, Monti, & Sandler, 2002) for a recent review. The earliest published papers in the field are due to Moorer (1977) and Piszczalski and Galler (1977). Moorer demonstrated a system that was capable of transcribing a limited polyphonic source such as a duet. Piszczalski and Galler (1977) focused on monophonic transcription. Their method analyses the music signal frame by frame. For each frame, they measure the fundamental frequency directly from local maxima of the Fourier transform magnitude. In this respect, this method is the first example of many other techniques that operate on a time-frequency distribution to estimate the fundamental frequency. Maher (1990) describes the first well-documented model in the literature that could track duets from real recordings by representing the audio signal as the superposition of sinusoids, known in the signal processing community as McAuley-Quatieri (MQ) analysis

(1986). Mellinger (1991) employed a cochleagram representation (a time-scale representation based on an auditory model (Slaney, 1995)). He proposed a set of directional filters for extracting features from this representation. Recently, Klapuri et al. (2001) proposed an iterative schema that operates on the frequency spectrum. They estimate a single dominant pitch, remove it from the energy spectrum and reestimate recursively on the residual. They report that the system outperforms expert human transcribers on a chord identification task.

Other attempts have been made to incorporate low level (physical) or high level (musical structure and cognitive) information for the processing of musical signals. Rossi, Girolami, and Leca (1997) reported a system that is based on matched filters estimated from piano sounds for polyphonic pitch identification for piano music. Martin (1999) has demonstrated use of a “blackboard architecture” (Klassner, Lesser, & Nawab, 1998; Mani, 1999) to transcribe polyphonic piano music (Bach chorales), that contained at most four different voices (bass-tenor-alto-soprano) simultaneously. Essentially, this is an expert system that encodes prior knowledge about physical sound characteristics, auditory physiology and high level musical structure such as rules of harmony. This direction is further exploited by (Bello, 2003). Good results reported by Rossi et al., Martin and Bello support the intuitive claim that combining prior information from both lower and higher levels can be very useful for transcription of musical signals.

In speech processing, tracking the pitch of a single speaker is a fundamental problem and methods proposed in the literature fill many volumes (Rabiner, Chen, Rosenberg, & McGonegal, 1976; Hess, 1983). Many of these techniques can readily be applied to monophonic music signals (de la Cuadra, Master, & Sapp, 2001; de Cheveigné & Kawahara, 2002). A closely related research effort to transcription is developing real-time pitch tracking and score following methods for interactive performance systems (Vercoe, 1984), or for fast sound to MIDI conversion (Lane, 1990). Score following applications can also be considered as pitch trackers with a very informative prior (i.e. they know what to look for). In such a context, Grubb (1998) developed a system that can track a vocalist given a score. A vast majority of pitch detection algorithms are based on heuristics (e.g., picking high energy peaks of a spectrogram, correlogram, auditory filter bank, e.t.c.) and their formulation usually lacks an explicit objective function or an explicit model. Hence, it is often difficult to theoretically justify merits and shortcomings of a proposed algorithm, compare it objectively to alternatives or extend it to more complex scenarios such as polyphony.

Pitch tracking is inherently related to detection and estimation of sinusoidals. Estimation and tracking of single or multiple sinusoidals is a fundamental problem in many branches of applied sciences so it is less surprising that the topic has also been deeply investigated in statistics, (e.g. see Quinn & Hannan, 2001). However, ideas from statistics seem to be not widely applied in the context of musical sound analysis, with only a few exceptions (Irizarry, 2001, 2002) who present frequentist techniques for very detailed analysis of musical sounds with particular focus on decomposition of periodic and transient components. (Saul, Lee, Isbell, & LeCun, 2002) presented real-time monophonic pitch tracking application based on Laplace approximation to the posterior parameter distribution of a second order autoregressive process (AR(2)) model (Truong-Van, 1990; Quinn & Hannan, 2001, page 19). Their method, with some rather simple preprocessing, outperforms several standard pitch tracking algorithms for speech, suggesting potential practical benefits of an approximate Bayesian treatment. For monophonic speech, a Kalman filter based pitch tracker is proposed by Parra and Jain (2001) that tracks parameters of a harmonic plus noise model (HNM). They propose the use of Laplace approximation around the predicted mean instead of the extended Kalman filter (EKF).

Statistical techniques have been applied for polyphonic transcription. Kashino is, to our knowledge, the first author to apply graphical models explicitly to the problem of music transcription. In Kashino et al. (1995), they construct a model to represent higher level musical knowledge and solve pitch identification separately. Sterian (1999) described a system that viewed transcription as a model driven segmentation of a time-frequency distribution. They use a Kalman filter model

to track partials on this image. Walmsley (2000) treats transcription and source separation in a full Bayesian framework. He employs a frame based generalized linear model (a sinusoidal model) and proposes a reversible-jump Markov Chain Monte Carlo (MCMC) (Andrieu & Doucet, 1999) inference algorithm. A very attractive feature of the model is that it does not make strong assumptions about the signal generation mechanism, and views the number of sources as well as the number of harmonics as unknown model parameters. Davy and Godsill (2003) address some of the shortcomings of his model and allow changing amplitudes and deviations in frequencies of partials from integer ratios. The reported results are good, however the method is computationally expensive. In a faster method, (Raphael, 2002) uses the short time Fourier Transform to make features and uses an HMM to infer most likely chord hypothesis.

In machine learning community, probabilistic models are widely applied for source separation, a.k.a. blind deconvolution, independent components analysis (ICA) (Hyvärinen, Karhunen, & Oja, 2001). Related techniques for source separation in music are investigated by (Casey, 1998). ICA models attempt source separation by forcing a factorized hidden state distribution, which can be interpreted as a “not-very-informative” prior. Therefore one needs typically multiple sensors for source separation. When the prior is more informative, one can attempt separation even from a single channel (Roweis, 2001; Jang & Lee, 2002; Hu & Wang, 2001).

Most of the authors view automated music transcription as a “audio to piano-roll” conversion and usually view “piano-roll to score” as a separate problem. This view is partially justified, since source separation and transcription from a polyphonic source is already a challenging task. On the other hand, automated generation of a human readable score includes nontrivial tasks such as tempo tracking, rhythm quantization, meter and key induction (Raphael, 2001a; Temperley, 2001). We argue that models described in this thesis allow for principled integration of higher level symbolic prior knowledge with low level signal analysis. Such an approach can guide and potentially improve the inference of a score, both in terms of quality of the solution and computation time.

1.3 Probabilistic Modelling and Music Transcription

We view music transcription, in particular rhythm quantization, tempo tracking and polyphonic pitch identification, as latent state estimation problems. In rhythm quantization or tempo tracking, given a sequence of onsets, we identify the most likely score or tempo trajectory. In polyphonic pitch identification, given the audio samples, we infer a piano-roll that represents the onset times, note durations and the pitch classes of individual notes.

Our general approach considers the quantities we wish to infer as a sequence of ‘hidden’ variables, which we denote simply by x . For each problem, we define a probability model, that relates the observations sequence y to the hidden x , possibly using a set of parameters θ . Given the observations, transcription can be viewed as a Bayesian inference problem, where we compute a posterior distribution over hidden quantities by “inverting” the model using the Bayes theorem.

1.3.1 Bayesian Inference

In Bayesian statistics, probability models are viewed as data structures that represent a model builders knowledge about a (possibly uncertain) phenomenon. The central quantity is a joint probability distribution:

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

that relates unknown variables x and unknown parameters θ to observations y . In probabilistic modelling, there is no fundamental difference between unknown variables and unknown model

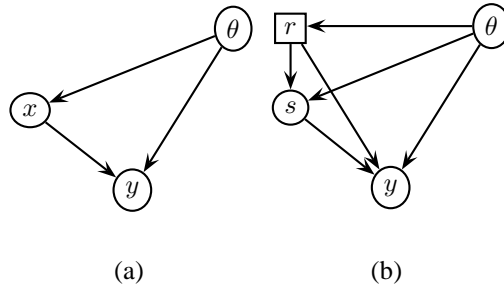


Figure 1.5: (a) Directed graphical model showing the assumed causal relationship between observables y , hidden variables x and parameters θ . (b) The hidden variables are further partitioned as $x = (s, r)$. Square nodes denote discrete, oval nodes denote continuous variables.

parameters; all can be viewed as unknown quantities to be estimated. The inference problem is to compute the posterior distribution using the Bayes theorem:

$$p(x, \theta|y) = \frac{1}{p(y)}p(y|\theta, x)p(x, \theta) \quad (1.1)$$

The prior term $p(x, \theta)$ reflects our knowledge about the parameters θ and hidden variables x before we observe any data. The likelihood model $p(y|\theta, x)$ relates θ and x to the observations y . It is usually convenient to think of $p(y|\theta, x)$ as a generative model for y . The model can be represented as a graphical model shown in Figure 1.5(a). Given the observations y , the posterior $p(x, \theta|y)$ reflects our entire knowledge (e.g., the probable values and the associated uncertainties) about the unknown quantities. A posterior distribution on the hidden variables can be obtained by integrating the joint posterior over the parameters, i.e.

$$p(x|y) = \int d\theta p(x, \theta|y) \quad (1.2)$$

From this quantity, we can obtain the most probable x^* given y as

$$x^* = \operatorname{argmax}_x p(x|y) \quad (1.3)$$

Unfortunately, the required integrations on θ are in most cases intractable so one has to resort to numerical or analytical approximation techniques. At this point, it is often more convenient to distinguish between x and θ to simplify approximations. For example, one common approach to approximation is to use a point estimate of the parameter and to convert intractable integration to a simple function evaluation. Such an estimate is the maximum a-posteriori (MAP) estimate given as:

$$\begin{aligned} \theta^* &= \operatorname{argmax}_\theta \int dx p(x, \theta|y) \\ p(x|y) &\approx p(x, \theta^*|y) \end{aligned}$$

Note that this formulation is equivalent to “learning” the best parameters given the observations. In some special cases, the required integrations over θ may still be carried out exactly. This includes the cases when y , x and θ are jointly Gaussian, or when both x and θ are discrete. Here, exact calculation hinges whether it is possible to represent the posterior $p(x, \theta|y)$ in a factorized form

using a data structure such as the *junction tree* (See (Smyth, Heckerman, & Jordan, 1996) and references herein).

Another source of intractability is reflected in combinatorial explosion. In some special hybrid model classes (such as switching linear dynamical systems (Murphy, 1998; Lerner & Parr, 2001)), we can divide the hidden variables in two sets $x = (s, r)$ where r is discrete and s given r is conditionally Gaussian (See Figure. 1.5(b)). We will use such models extensively in the thesis. To infer the most likely r consistent with the observations, we need to compute

$$r^* = \operatorname{argmax}_r \int ds d\theta p(r, s, \theta | y)$$

If we assume that model parameters θ are known, (e.g. suppose we have estimated θ^* on a training set where r was known) we can simplify the problem as:

$$r^* \approx \operatorname{argmax}_r p(r | y) = \operatorname{argmax}_r \int ds p(y | r, s) p(s | r) p(r) \quad (1.4)$$

Here, we have omitted explicit conditioning on θ^* . We can evaluate the integral in Eq.1.4 for any given r . However, in order to find the optimal solution r^* exactly, we still need to evaluate the the integral separately for every r in the configuration space. Apart from some special cases, where we can derive exact polynomial time algorithms; in general the only exact method is exhaustive search. Fortunately, although finding r^* is intractable in general, in practice a useful solution may be found by approximate methods. Intuitively, this is due to fact that realistic priors $p(r)$ are usually very informative (most of the configurations r have very small probability) and the likelihood term $p(y | r)$ is quite crisp. All this factors tend to render the posterior unimodal.

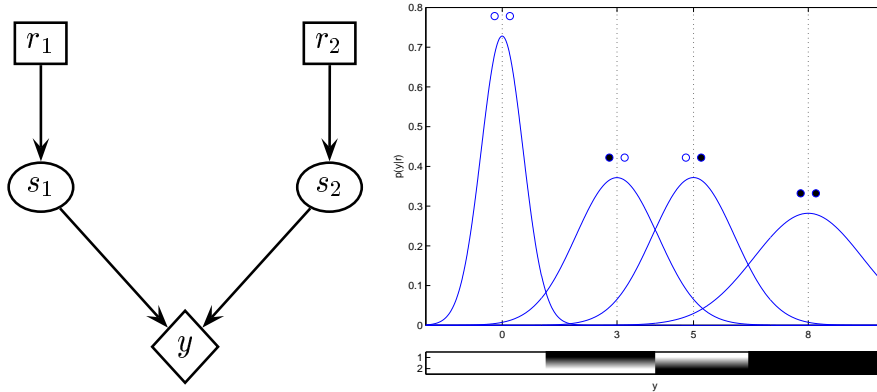
1.4 A toy example

We will now illustrate the basic ideas of Bayesian inference developed in the previous section on a toy sequencer model. The sequencer model is quite simple and is able to generate output signals of length one only. We denote this output signal as y . The “scores” that it can process are equally limited and can consist of at most two “notes”. Hence, the “musical universe” of our sequencer is limited only to 4 possible scores, namely *silence*, two single note melodies and one two note chord. Given any one of the four possible scores, the sequencer generates control signals which we will call a “piano-roll”. In this representation, we will encode each note by a bit $r_j \in \{\text{“sound”}, \text{“mute”}\}$ for $j = 1, 2$. This indicator bit denotes simply whether the j ’th note is present in the score or not. In this simplistic example, there is no distinction between a score and a piano-roll and the latter is merely an encoding of the former; but for longer signals there will be a distinction. We specify next what waveform the sequencer should generate when a note is present or absent. We will denote this waveform by s_j

$$s_j | r_j \sim [r_j = \text{sound}] \mathcal{N}(s_j; \mu_j, P_s) + [r_j = \text{mute}] \mathcal{N}(s_j; 0, P_m)$$

Here the notation $[x = \text{text}]$ has value equal to 1 when variable x is in state *text*, and is zero otherwise. The symbol $\mathcal{N}(s; \mu, P)$ denotes a Gaussian distribution on variable s with mean μ and variance P . Verbally, the above equation means that when $r_j = \text{mute}$, $s_j \approx 0 \pm \sqrt{P_m}$ and when $r_j = \text{sound}$, $s_j \approx \mu_j \pm \sqrt{P_s}$. Here the μ_j , P_s and P_m are known parameters of the signal model. Finally, the output signal is given by summing up each waveform of individual notes

$$y = \sum_j s_j$$



(a) Graphical model of the toy sequencer model. Square and oval shaped nodes denote discrete (piano-roll) and continuous (waveform) variables respectively. Diamond-shaped node represents the observed signal.

(b) The conditional $p(y|r_1, r_2)$. The “mute” and “sound” states are denoted by \circ and \bullet respectively. Here, $\mu_1 = 3$, $\mu_2 = 5$ and $P_m < P_s$. The bottom figure shows the most likely transcription as a function of y , i.e. $\arg \max_{r_1, r_2} p(r_1, r_2|y)$. We assume a flat prior, $p(r_j = \text{“mute”}) = p(r_j = \text{“sound”}) = 0.5$.

Figure 1.6: Graphical model for the toy sequencer model

To make the model complete, we have to specify a prior distribution that describes how the scores are generated. Since there is no distinction between a piano-roll and a score in this example, we will directly define a prior directly on piano-roll. For simplicity, we assume that notes are a-priori independent, i.e.

$$r_j \sim p(r_j) \quad j = 1, 2$$

and choose a uniform prior with $p(r_j = \text{mute}) = p(r_j = \text{sound}) = 0.5$. The corresponding graphical model for this generative process is shown in Figure 1.6.

The main role of the generative process is that it makes it conceptually easy to describe a joint distribution between the output signal y , waveforms $\mathbf{s} = (s_1, s_2)$ and piano-roll $\mathbf{r} = (r_1, r_2)$ where

$$p(y, \mathbf{s}, \mathbf{r}) = p(y|\mathbf{s})p(\mathbf{s}|\mathbf{r})p(\mathbf{r})$$

Moreover, this construction implies a certain *factorization* which potentially simplifies both the representation of the joint distribution and the inference procedure. Formally, the transcription task is now to calculate the conditional probability which is given by the *Bayes theorem* as

$$p(\mathbf{r}|y) = \frac{1}{p(y)} p(y|\mathbf{r})p(\mathbf{r})$$

Here, $p(y) = \sum_{\mathbf{r}} p(y|\mathbf{r})p(\mathbf{r})$ is a normalization constant. In transcription, we are interested into the most likely piano-roll \mathbf{r}^* , hence the actual numerical value $p(y)$, which merely scales the objective, is at this point not important, i.e. we have

$$\mathbf{r}^* = \underset{\mathbf{r}}{\operatorname{argmax}} p(\mathbf{r}|y) = \underset{\mathbf{r}}{\operatorname{argmax}} p(y|\mathbf{r})p(\mathbf{r}) \quad (1.5)$$

The prior factor $p(\mathbf{r})$ is already specified. The other term can be calculated by *integrating out* the waveforms \mathbf{s} , i.e.

$$p(y|\mathbf{r}) = \int d\mathbf{s} p(y, \mathbf{s}|\mathbf{r}) = \int d\mathbf{s} p(y|\mathbf{s})p(\mathbf{s}|\mathbf{r})$$

Conditioned on any \mathbf{r} , this quantity can be found analytically. For example, when $r_1 = r_2 =$ “sound”, $p(y|\mathbf{r}) = \mathcal{N}(y; \mu_1 + \mu_2, 2P_s)$. A numeric example is shown in Figure 1.6.

This simple toy example exhibits the key idea in our approach. Basically, by just carefully describing the sound generation procedure, we were able to formulate an optimization problem (Eq. 1.5) for doing polyphonic transcription! The derivation is entirely mechanical and ensures that the objective function consistently incorporates our prior knowledge about scores and about the sound generation procedure (through $p(\mathbf{r})$ and $p(\mathbf{s}|\mathbf{r})$). Of course, in reality, y and each of r_j and s_j will be time series and both the score and sound generation process will be far more complex. But most importantly, we have divided the problem into two parts, in one part formulating a realistic model, on the other part finding an efficient inference algorithm.

1.5 Outline of the thesis

In the following chapters, we describe several methods for transcription. For each subproblem, we define a probability model, that relates the observations, hidden and parameters. The particular definition of these quantities will depend on the context, but observables and hidden will be sequences of random variables. For a given observation sequence, we will compute the posterior distribution or some posterior features such as the MAP.

In Chapter 2, we describe a model that relates short scores with corresponding onset times of events in an expressive performance. The parameters of the model is trained on data resulting from a psychoacoustical experiment to mimic the behaviour of a human transcriber on this task. This chapter addresses the issue that there is not a single “ground truth” in music transcription. Even for very simple rhythms, well trained human subjects show significant variations in their responses. We demonstrate how this uncertainty problem can be addressed naturally using a probabilistic model.

Chapter 3 focuses on tempo tracking from onsets. The observation model is a multiscale representation (analogous to a wavelet transform). The tempo prior is modelled as a Gauss-Markov process. The tempo is viewed as a hidden state variable and is estimated by approximate Kalman filtering.

We introduce in Chapter 4 a generative model to combine rhythm quantization and tempo tracking. The model is a switching state space model in which computation of exact probabilities becomes intractable. We introduce approximation techniques based on simulation, namely Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC).

In Chapter 5, we propose a generative model for polyphonic transcription from audio signals. The model, formulated as a Dynamical Bayesian Network, describes the relationship between polyphonic audio signal and an underlying piano roll. This model is also a special case of the, generally intractable, switching state space model. Where possible, we derive, exact polynomial time inference procedures, and otherwise efficient approximations.

1.6 Future Directions and Conclusions

When transcribing music, human experts rely heavily on prior knowledge about the musical structure – harmony, tempo, timbre, expression, e.t.c. As partially demonstrated in this thesis and elsewhere (e.g. (Raphael & Stoddard, 2003)), such structure can be captured by training probabilistic

generative models on a corpus of example compositions, performances or sounds by collecting statistics over selected features. One of the important advantages of our approach is that, at least in principle, prior knowledge about any type of musical structure can be consistently integrated. An attempt in this direction is made in (Cemgil, Kappen, & Barber, 2003), where we described a model that combines low level signal analysis with high level knowledge. However, the computational obstacles and software engineering issues are yet to be overcome. I believe that investigation of this direction is important in designing robust and practical music transcription systems.

In my view, the most attractive feature of probabilistic modelling and Bayesian inference for music transcription is the decoupling of modelling from inference. In this framework, the model clearly describes the objective and the question how we actually solve the objective, whilst equally important, becomes an entirely algorithmic and computational issue. Particularly in music transcription, as in many other perceptual tasks, the answer to the question of “what to optimize” is far from trivial. This thesis tries to answer this question by defining an objective by using probabilistic generative models and touches upon some state-of-the-art inference techniques for its solution.

I argue that practical polyphonic music transcription can be made computationally easy; the difficulty of the problem lies in formulating precisely what the objective is. This is in contrast with traditional problems of computer science, such as the travelling salesman problem, which are very easy to formulate but difficult to solve exactly. In my view, this fundamental difference in the nature of the music transcription problem requires a model-centred approach rather than an algorithm-centred approach. One can argue that objectives formulated in the context of probabilistic models are often intractable. I answer this by paraphrasing John Tukey, who in the 50’s said “An approximate solution of the exact problem is often more useful than the exact solution of an approximate problem”.

Chapter 2

Rhythm Quantization

One important task in music transcription is rhythm quantization that refers to categorization of note durations. Although quantization of a pure mechanical performance is rather straightforward, the task becomes increasingly difficult in presence of musical expression, i.e. systematic variations in timing of notes and in tempo. In this chapter, we assume that the tempo is known. Expressive deviations are modelled by a probabilistic performance model from which the corresponding optimal quantizer is derived by Bayes theorem. We demonstrate that many different quantization schemata can be derived in this framework by proposing suitable prior and likelihood distributions. The derived quantizer operates on short groups of onsets and is thus flexible both in capturing the structure of timing deviations and in controlling the complexity of resulting notations. The model is trained on data resulting from a psychoacoustical experiment and thus can mimic the behaviour of a human transcriber on this task.

Adapted from A.T. Cemgil, P. Desain, and H.J. Kappen. Rhythm quantization for transcription. *Computer Music Journal*, pages 60–75, 2000.

2.1 Introduction

One important task in music transcription is rhythm quantization that refers to categorization of note durations. Quantization of a “mechanical” performance is rather straightforward. On the other hand, the task becomes increasingly difficult in presence of expressive variations, that can be thought as systematic deviations from a pure mechanical performance. In such unconstrained performance conditions, mainly two types of systematic deviations from exact values do occur. At small time scale notes can be played accented or delayed. At large scale tempo can vary, for example the musician(s) can accelerate (or decelerate) during performance or slow down (ritard) at the end of the piece. In any case, these timing variations usually obey a certain structure since they are mostly intended by the performer. Moreover, they are linked to several attributes of the performance such as meter, phrase, form, style etc. (Clarke, 1985). To devise a general computational model (i.e. a performance model) which takes all these factors into account, seems to be quite hard.

Another observation important for quantization is that we perceive a rhythmic pattern not as a sequence of isolated onsets but rather as a perceptual entity made of onsets. This also suggests that attributes of neighboring onsets such as duration, timing deviation etc. are correlated in some way.

This correlation structure is not fully exploited in commercial music packages, which do automated music transcription and score type setting. The usual approach taken is to assume a constant tempo throughout the piece, and to quantize each onset to the nearest grid point implied by the tempo and a suitable pre-specified minimum note duration (e.g. eight, sixteenth etc.). Such a grid

quantization schema implies that each onset is quantized to the nearest grid point *independent* of its neighbours and thus all of its attributes are assumed to be independent, hence the correlation structure is not employed. The consequence of this restriction is that users are required to play along with a fixed metronome and without any expression. The quality of the resulting quantization is only satisfactory if the music is performed according to the assumptions made by the quantization algorithm. In the case of grid-quantization this is a mechanical performance with small and independent random deviations.

More elaborate models for rhythm quantization indirectly take the correlation structure of expressive deviations into account. In one of the first attempt to quantization, (Longuet-Higgins, 1987) described a method in which he uses hierarchical structure of musical rhythms to do quantization. (Desain, Honing, & de Rijk, 1992) use a relaxation network in which pairs of time intervals are attracted to simple integer ratios. (Pressing & Lawrence, 1993) use several template grids and compare both onsets and inter-onset intervals (IOI's) to the grid and select the best quantization according to some distance criterion. The Kant system (Agon et al., 1994) developed at IRCAM uses more sophisticated heuristics but is in principle similar to (Pressing & Lawrence, 1993).

The common critic to all of these models is that the assumptions about the expressive deviations are implicit and are usually hidden in the model, thus it is not always clear how a particular design choice effects the overall performance for a full range of musical styles. Moreover it is not directly possible to use experimental data to tune model parameters to enhance the quantization performance.

In this chapter, we describe a method for quantization of onset sequences. The paper is organized as follows: First, we state the transcription problem and define the terminology. Using the Bayesian framework we briefly introduce, we describe probabilistic models for expressive deviation and notation complexity and show how different quantizers can be derived from them. Consequently, we train the resulting model on experimental data obtained from a psychoacoustical experiment and compare its performance to simple quantization strategies.

2.2 Rhythm Quantization Problem

2.2.1 Definitions

A *performed rhythm* is denoted by a sequence $[t_i]$ ¹ where each entry is the time of occurrence of an onset. For example, the performed rhythm in Figure 1.3(a) is represented by $t_1 = 0, t_2 = 1.18, t_3 = 1.77, t_4 = 2.06$ etc. We will also use the terms *performance* or *rhythm* interchangeably when we refer to an onset sequence.

A very important subtask in transcription is tempo tracking, i.e. the induction of a sequence of points (i.e. *beats*) in time, which coincides with the human sense of rhythm (e.g. foot tapping) when listening to music. We call such a sequence of beats a *tempo track* and denote it by $\vec{\tau} = [\tau_j]$ where τ_j is the time at which j 'th beat occurs. We note that for automatic transcription, $\vec{\tau}$ is to be estimated from $[t_i]$.

Once a tempo track $\vec{\tau}$ is given, the rhythm can be segmented into a sequence of segments, each of duration $\tau_j - \tau_{j-1}$. The j 'th segment will contain K_j onsets, which we enumerate by $k = 1 \dots K_j$. The onsets in each segment are normalized and denoted by $\mathbf{t}_j = [t_j^k]$, i.e. for all $\tau_{j-1} \leq t_i < \tau_j$ where

$$t_j^k = \frac{t_i - \tau_{j-1}}{\tau_j - \tau_{j-1}} \quad (2.1)$$

¹We will denote a set with the typical element x_j as $\{x_j\}$. If the elements are ordered (e.g. to form a vector) we will use $[x_j]$.

Note that this is merely a reindexing from single index i to double index (k, j) ². In other words the onsets are scaled and translated such that an onset just at the end of the segment is mapped to one and another just at the beginning to zero. The segmentation of a performance is given in Figure 2.1.

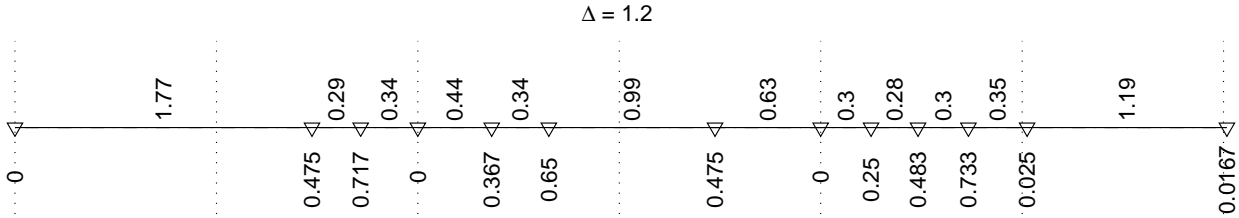


Figure 2.1: Segmentation of a performance by a tempo track (vertical dashed lines) $\vec{\tau} = [0.0, 1.2, 2.4, 3.6, 4.8, 6.0, 7.2, 8.4]$. The resulting segments are $t_0 = [0]$, $t_1 = [0.475, 0.717]$ etc.

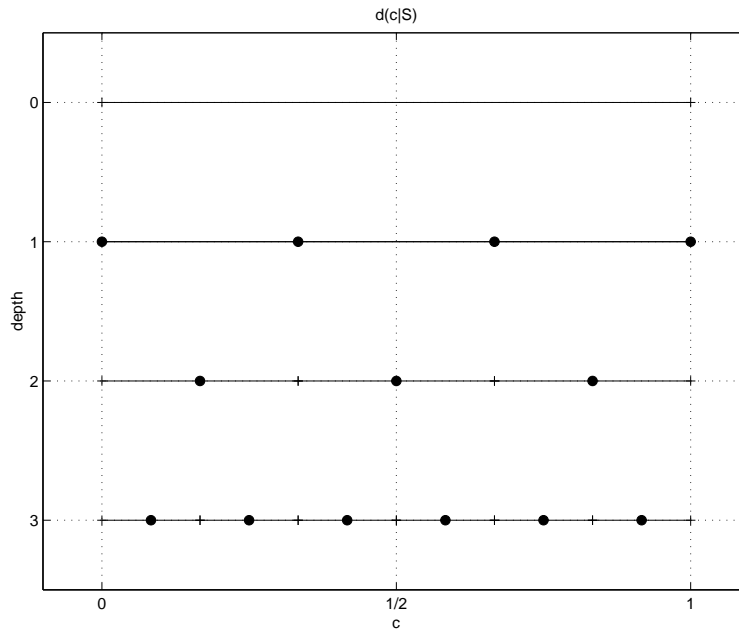


Figure 2.2: Depth of gridpoint c by subdivision schema $\mathcal{S} = [3, 2, 2]$

Once a segmentation is given, quantization reduces to mapping onsets to locations, which can be described by simple rational numbers. Since in western music tradition, notations are generated by recursive subdivisions of a whole note, it is also convenient to generate possible onset quantization locations by regular subdivisions. We let $\mathcal{S} = [s_i]$ denote a subdivision schema, where $[s_i]$ is a sequence of small prime numbers. Possible quantization locations are generated by subdividing the unit interval $[0, 1]$. At each new iteration i , the intervals already generated are divided further into s_i equal parts and the resulting endpoints are added to a set C . Note that this procedure places the quantization locations on a grid of points c_n where two neighboring grid points have the distance $1 / \prod_i s_i$. We will denote the first iteration number at which the grid point c is added to C as the *depth* of c with respect to \mathcal{S} . This number will be denoted as $d(c|\mathcal{S})$.

As an example consider the subdivision $\mathcal{S} = [3, 2, 2]$. The unit interval is divided first into three equal pieces, then the resulting intervals into 2 and etc. At each iteration, generated endpoints are

²When an argument applies to all segments, we will drop the index j .

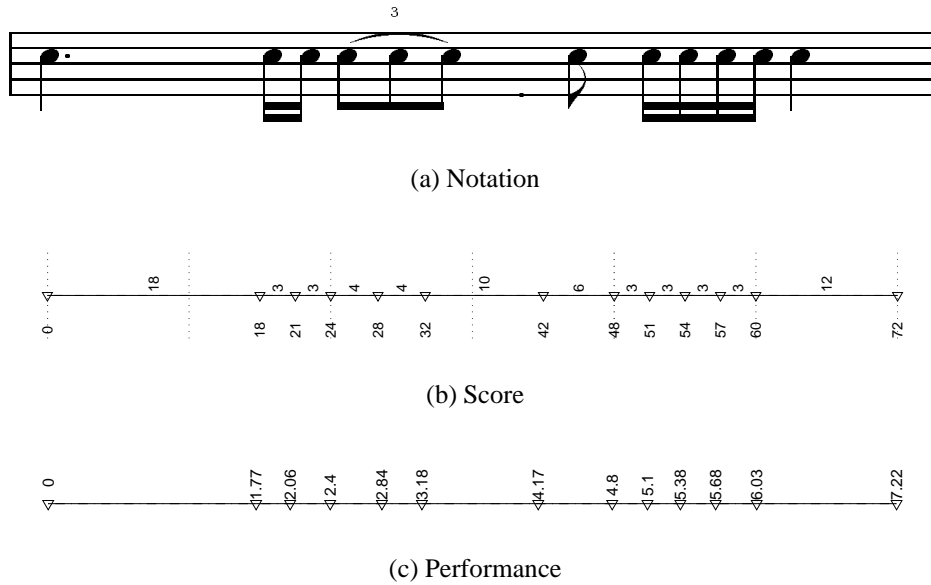


Figure 2.3: A simplified schema of onset quantization. A notation (a) defines a score (b) which places onsets on simple rational points with respect to a tempo track (vertical dashed lines). The performer “maps” (b) to a performance (c). This process is not deterministic; in every new performance of this score a (slightly) different performance would result. A performance model is a description of this stochastic process. The task of the transcriber is to recover both the tempo track and the onset locations in (b) given (c).

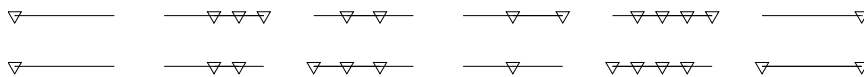


Figure 2.4: Two equivalent representations of the notation in Figure 2.3(a) by a code vector sequence. Here, each horizontal line segment represents one vector of length one beat. The endpoint of one vector is the same point in time as the beginning of the next vector. Note that the only difference between two equivalent representations is that some begin and endpoints are swapped.

added to the list. In the first iteration, 0, 1/3, 2/3 and 1 are added to the list. In the second iteration, 1/6, 3/6 and 5/6 are added, etc. The resulting grid points (filled circles) are depicted in Figure 2.2. The vertical axis corresponds to $d(c|\mathcal{S})$.

If a segment \mathbf{t} is quantized (with respect to \mathcal{S}), the result is a K dimensional vector with all entries on some grid points. Such a vector we call a *code vector* and denote as $\mathbf{c} = [c_k]$, i.e. $\mathbf{c} \in C \times C \cdots \times C = C^K$. We call a set of code-vectors a *codebook*. Since all entries of a code vector coincide with some grid points, we can define the *depth of a code vector* as

$$d(\mathbf{c}|\mathcal{S}) = \sum_{c_k \in \mathbf{c}} d(c_k|\mathcal{S}) \tag{2.2}$$

A score can be viewed as a *concatenation* of code vectors \mathbf{c}_j . For example, the notation in Figure 2.3(a) can be represented by a code vector sequence as in Figure 2.4. Note that the representation is not unique, both code vector sequences represent the same notation.

2.2.2 Performance Model

As described in the introduction section, natural music performance is subject to several systematic deviations. In lack of such deviations, every score would have only one possible interpretation. Clearly, two natural performances of a piece of music are never the same, even performance of very short rhythms show deviations from a strict mechanical performance. In general terms, a *performance model* is a mathematical description of such deviations, i.e. it describes how likely it is that a score is mapped into a performance (Figure 2.3). Before we describe a probabilistic performance model, we briefly review a basic theorem of probability theory.

2.2.3 Bayes Theorem

The joint probability $p(A, B)$ of two random variables A and B defined over the respective state spaces S_A and S_B can be factorized in two ways:

$$p(A, B) = p(B|A)p(A) = p(A|B)p(B) \quad (2.3)$$

where $p(A|B)$ denotes the conditional probability of A given B : for each value of B , this is a probability distribution over A . Therefore $\sum_A p(A|B) = 1$ for any fixed B . The marginal distribution of a variable can be found from the joint distribution by summing over all states of the other variable, e.g.:

$$p(A) = \sum_{B \in S_B} p(A, B) = \sum_{B \in S_B} p(A|B)p(B) \quad (2.4)$$

It is understood that summation is to be replaced by integration if the state space is continuous. Bayes theorem results from Eq. 2.3 and Eq. 2.4 as:

$$p(B|A) = \frac{p(A|B)p(B)}{\sum_{B \in S_B} p(A|B)p(B)} \quad (2.5)$$

$$\propto p(A|B)p(B) \quad (2.6)$$

The proportionality follows from the fact that the denominator does not depend on B , since B is already summed over. This rather simple looking “formula” has surprisingly far reaching consequences and can be directly applied to quantization. Consider the case that B is a score and S_B is the set of all possible scores. Let A be the observed performance. Then Eq 2.5 can be written as

$$p(\text{Score}|\text{Performance}) \propto p(\text{Performance}|\text{Score}) \times p(\text{Score}) \quad (2.7)$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior} \quad (2.8)$$

The intuitive meaning of this equation can be better understood, if we think of quantization as a score selection problem. Since there is usually not a single true notation for a given performance, there will be several possibilities. The most reasonable choice is selecting the score \mathbf{c} which has the highest probability given the performance \mathbf{t} . Technically, we name this probability distribution as the posterior $p(\mathbf{c}|\mathbf{t})$. The name posterior comes from the fact that this quantity appears *after* we observe the performance \mathbf{t} . Note that the posterior is a function over \mathbf{c} , and assigns a number to each notation after we fix \mathbf{t} . We look for the notation \mathbf{c} that maximizes this function. Bayes theorem tells us that the posterior is proportional to the product of two quantities, the likelihood $p(\mathbf{t}|\mathbf{c})$ and the prior $p(\mathbf{c})$. Before we explain the interpretation of the likelihood and the prior in this context, we first summarize the ideas in compact notation as

$$p(\mathbf{c}|\mathbf{t}) \propto p(\mathbf{t}|\mathbf{c})p(\mathbf{c}). \quad (2.9)$$

The best code vector \mathbf{c}^* is given by

$$\mathbf{c}^* = \underset{\mathbf{c} \in \mathbf{C}^{\mathbf{K}}}{\operatorname{argmax}} p(\mathbf{c}|\mathbf{t}) \quad (2.10)$$

In technical terms, this problem is called a maximum a-posteriori (MAP) estimation problem and \mathbf{c}^* is called the MAP solution of this problem. We can also define a related quantity \mathcal{L} (minus log-posterior) and try to minimize this quantity rather than maximizing Eq. 2.9 directly. This simplifies the form of the objective function without changing the locations of local extrema since $\log(x)$ is a monotonically increasing function.

$$\mathcal{L} = -\log p(\mathbf{c}|\mathbf{t}) \propto -\log p(\mathbf{t}|\mathbf{c}) + \log \frac{1}{p(\mathbf{c})} \quad (2.11)$$

The $-\log p(\mathbf{t}|\mathbf{c})$ term in Equation 2.11, which is the minus logarithm of the likelihood, can be interpreted as a distance measuring how far the rhythm \mathbf{t} is played from the perfect mechanical performance \mathbf{c} . For example, if $p(t|c)$ would be of form $\exp(-(t-c)^2)$, then $-\log(t|c)$ would be $(t-c)^2$, the square of the distance from t to c . This quantity can be made arbitrary small if we use a very fine grid, however, as mentioned in the introduction section, this eventually would result in a complex notation. A suitable prior distribution prevents this undesired result. The $\log \frac{1}{p(\mathbf{c})}$ term, which is large when the prior probability $p(\mathbf{c})$ of the codevector is small, can be interpreted as a complexity term, which penalizes complex notations. The best quantization balances the likelihood and the prior in an optimal way. The precise form of the prior will be discussed in a later section.

The form of a performance model, i.e. the likelihood, can be in general very complicated. However, in this article we will consider a subclass of performance models where the expressive timing is assumed to be an additive noise component which depends on \mathbf{c} . The model is given by

$$\mathbf{t}_j = \mathbf{c}_j + \varepsilon_j \quad (2.12)$$

where ε_j is a vector which denotes the *expressive timing deviation*. In this paper we will assume that ε is normal distributed with zero mean and covariance matrix $\Sigma_\varepsilon(\mathbf{c})$, i.e. the correlation structure depends upon the code vector. We denote this distribution as $\varepsilon \sim \mathcal{N}(0, \Sigma_\varepsilon(\mathbf{c}))$. Note that when ε is the zero vector, ($\Sigma_\varepsilon \rightarrow \mathbf{0}$), the model reduces to a so-called “mechanical” performance.

2.2.4 Example 1: Scalar Quantizer (Grid Quantizer)

We will now demonstrate on a simple example how these ideas are applied to quantization.

Consider a one-onset segment $\mathbf{t} = [0.45]$. Suppose we wish to quantize the onset to one of the endpoints, i.e. we are using effectively the codebook $\mathbf{C} = \{[0], [1]\}$. The obvious strategy is to quantize the onset to the nearest grid point (e.g. a grid quantizer) and so the code-vector $\mathbf{c} = [0]$ is chosen as the winner.

The Bayesian interpretation of this decision can be demonstrated by computing the corresponding likelihood $p(\mathbf{t}|\mathbf{c})$ and the prior $p(\mathbf{c})$. It is reasonable to assume that the probability of observing a performance \mathbf{t} given a particular \mathbf{c} decreases with the distance $|\mathbf{c} - \mathbf{t}|$. A probability distribution having this property is the normal distribution. Since there is only one onset, the dimension $K = 1$ and the likelihood is given by

$$p(t|c) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t-c)^2}{2\sigma^2}\right)$$

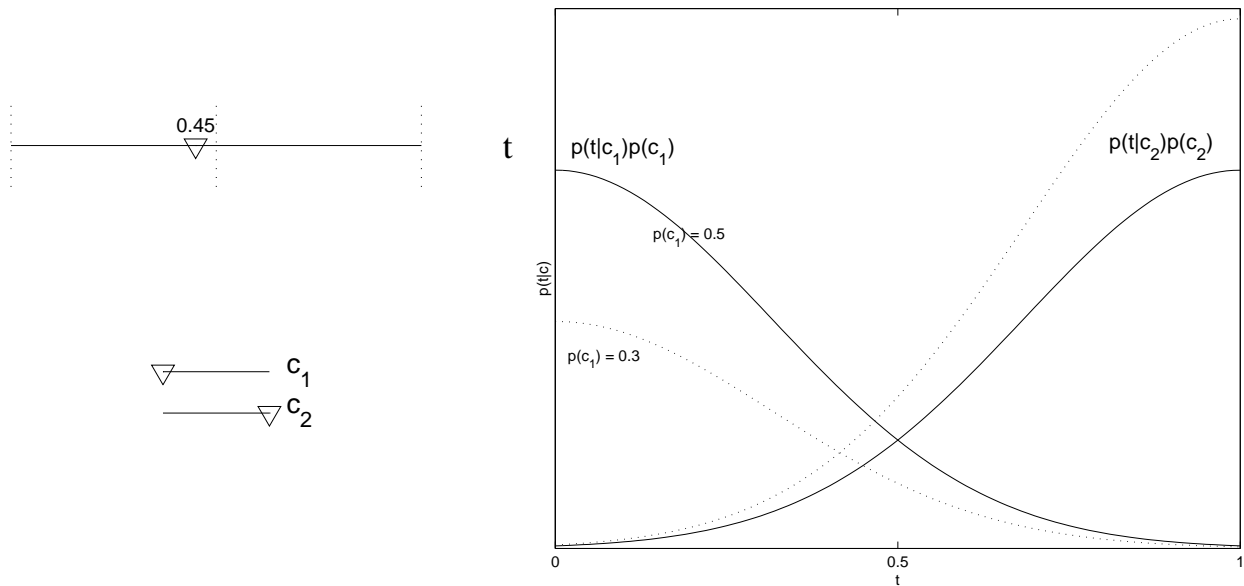


Figure 2.5: Quantization of an onset as Bayesian Inference. When $p(c) = [1/2, 1/2]$, at each t , the posterior $p(c|t)$ is proportional to the solid lines, and the decision boundary is at $t = 0.5$. When the prior is changed to $p(c) = [0.3, 0.7]$ (dashed), the decision boundary moves towards 0.

If both codevectors are equally probable, a flat prior can be chosen, i.e. $p(c) = [1/2, 1/2]$. The resulting posterior $p(c|t)$ is plotted in 2.5. The decision boundary is at $t = 0.5$, where $p(c_1|t) = p(c_2|t)$. The winner is given as in Eq. 2.10

$$c^* = \underset{c}{\operatorname{argmax}} p(c|t)$$

Different quantization strategies can be implemented by changing the prior. For example if $c = [0]$ is assumed to be less probable, we can choose another prior, e.g. $p(c) = [0.3, 0.7]$. In this case the decision boundary shifts from 0.5 towards 0 as expected.

2.2.5 Example 2: Vector Quantizer

Assigning different prior probabilities to notations is only one way of implementing different quantization strategies. Further decision regions can be implemented by varying the conditional probability distribution $p(t|c)$. In this section we will demonstrate the flexibility of this approach for quantization of groups of onsets.

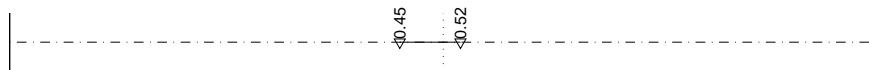


Figure 2.6: Two Onsets

Consider the segment $t = [0.45, 0.52]$ depicted in Figure 2.6. Suppose we wish to quantize the onsets again only to one of the endpoints, i.e. we are using effectively the codebook $C = \{[0, 0], [0, 1], [1, 1]\}$. The simplest strategy is to quantize every onset to the nearest grid point (e.g. a grid quantizer) and so the code-vector $c = [0, 1]$ is the winner. However, this result might be not very desirable, since the inter-onset interval (IOI) has increased more than 14 times, (from 0.07 to 1). It is less likely that a human transcriber would make this choice since it is perceptually not very

realistic. We could try to solve this problem by employing another strategy : If $\delta = t_2 - t_1 > 0.5$, we use the code-vector $[0, 1]$. If $\delta \leq 0.5$, we quantize to one of the code-vectors $[0, 0]$ or $[1, 1]$ depending upon the average of the onsets. In this strategy the quantization of $[0.45, 0.52]$ is $[0, 0]$.

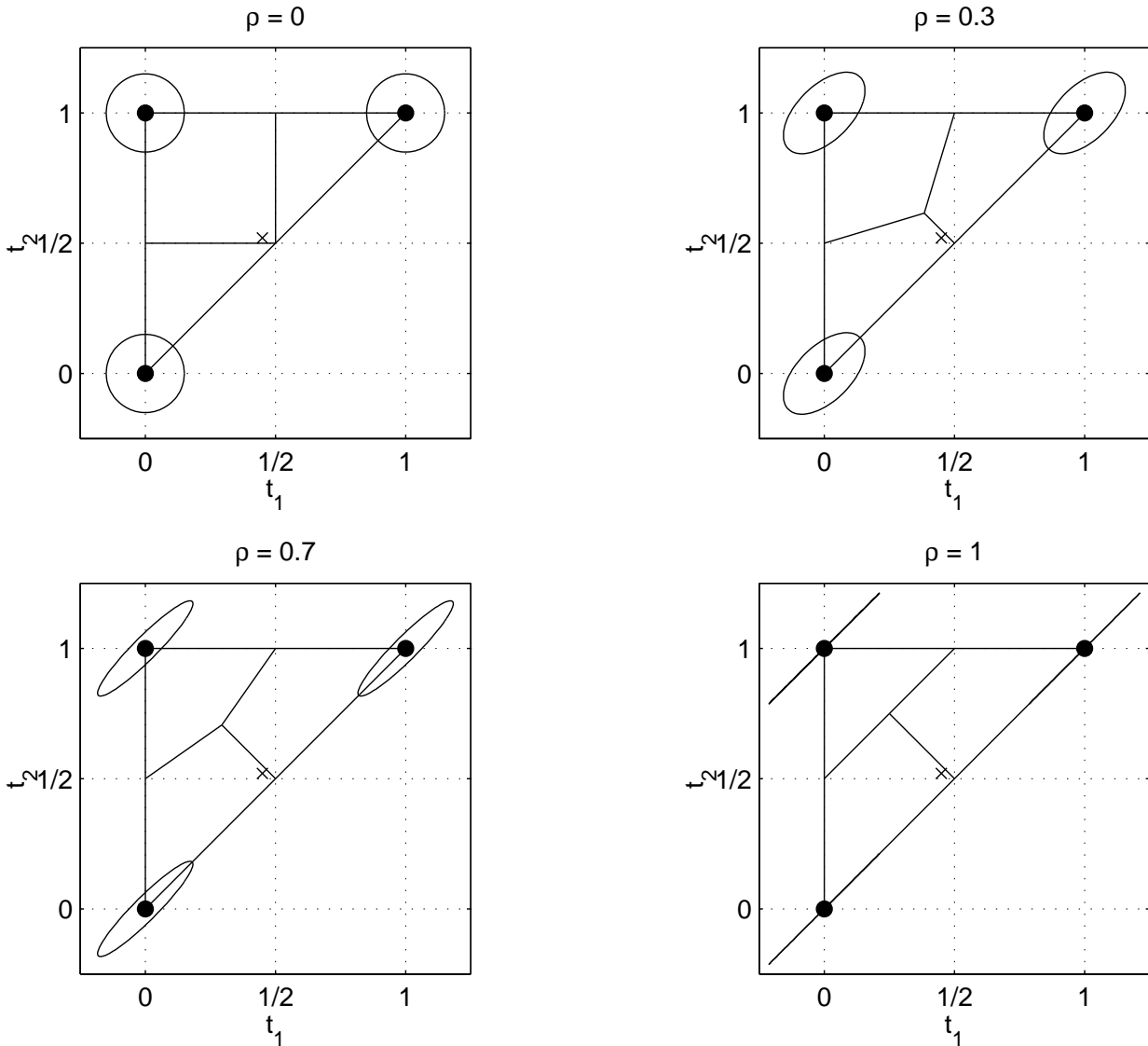


Figure 2.7: Tiling for choices of ρ and constant $p(c)$. Onset quantization (i.e. grid quantization) used by many commercial notation packages corresponds to the case where $\rho = 0$. IOI quantization appears when $\rho \rightarrow 1$. Note that different correlation structures imply different quantization decisions, not necessarily onset- or IOI-quantization. The cross corresponds to the rhythm $\mathbf{t} = [0.45, 0.52]$.

Although considered to be different in the literature, both strategies are just special cases which can be derived from Eq. 2.11 by making specific choices about the correlation structure (covariance matrix Σ_ε) of expressive deviations. The first strategy assumes that the expressive deviations of both onsets are independent of each other. This is apparently not a very realistic model for timing deviations in music. The latter corresponds to the case where onsets are linearly dependent; it was assumed that $t_2 = t_1 + \delta$ and only δ and t_1 were considered in quantization. This latter operation is merely a linear transformation of onset times and is implied by the implicit assumption about

the correlation structure. Indeed some quantization models in the literature focus directly on IOI's rather than on onset times.

More general strategies, which can be quite difficult to state verbally, can be specified by different choices of Σ_ε and $p(\mathbf{c})$. Some examples for the choice $\Sigma_\varepsilon = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ and constant $p(\mathbf{c})$ are depicted in Figure 2.7. The ellipses denote the set of points which are equidistant from the center and the covariance matrix Σ_ε determines their orientation. The lines denote the decision boundaries. The interested reader is referred to (Duda & Hart, 1973) for a discussion of the underlying theory.

Likelihood for the Vector Quantizer

For modeling the expressive timing ε in a segment containing K onsets, we propose the following parametric form for the covariance matrix

$$\Sigma_\varepsilon(\mathbf{c}) = \sigma^2 \begin{pmatrix} 1 & \rho_{1,2} & \cdots & \rho_{1,K} \\ \rho_{1,2} & 1 & \rho_{n,m} & \vdots \\ \vdots & \rho_{n,m} & \ddots & \vdots \\ \rho_{1,K} & \cdots & \cdots & 1 \end{pmatrix} \quad (2.13)$$

where

$$\rho_{n,m} = \eta \exp\left(-\frac{\lambda^2}{2}(c_m - c_n)^2\right) \quad (2.14)$$

Here, c_m and c_n are two distinct entries (grid points) of the code vector \mathbf{c} . In Eq. 2.14, η is a parameter between -1 and 1, which adjust the amount of correlation strength between two onsets. The other parameter λ adjusts the correlation as a function of the distance between entries in the code vector. When λ is zero, all entries are correlated by the equal amount, namely η . When λ is large, the correlation approaches rapidly to zero with increasing distance.

This particular choice for $p(\varepsilon)$ reflects the observation that onsets, which are close to each other, tend to be highly correlated. This can be interpreted as follows: if the onsets are close to each other, it is easier to quantify the IOI and then select an appropriate translation for the onsets by keeping the IOI constant. If the grid points are far away from each other, the correlation tends to be weak (or sometimes negative), which suggests that onsets are quantized independently of each other. In section 2.3, we will verify this choice empirically.

Prior for the Vector Quantizer

The choice of the prior $p(\mathbf{c})$ reflects the complexity of codevector \mathbf{c} . In this article we propose a complexity measure from a probabilistic point of view. In this measure, the complexity of a codevector $\mathbf{c} = [c_i]$ is determined by the depth of c_i with respect to the beat (See Eq. 2.2) and the time signature of the piece. See Figure 2.8.

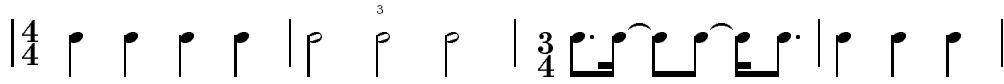
The prior probability of a code-vector with respect to \mathcal{S} is chosen as

$$p(\mathbf{c}|\mathcal{S}) \propto e^{-\gamma d(\mathbf{c}|\mathcal{S})} \quad (2.15)$$

Note that if $\gamma = 0$, then the depth of the codevector has no influence upon its complexity. If it is large, (e.g. $\gamma \approx 1$) only very simple rhythms get reasonable probability mass. practice, we choose $\gamma \approx 0.02$. This choice is also in accordance with the intuition and experimental evidence: simpler



(a) In lack of any other context, both onset sequences will sound the same. However the first notation is more complex



(b) Assumed time signature determines the complexity of a notation

Figure 2.8: Complexity of a notation

rhythms are more frequently used than complex ones. The marginal prior of a codevector is found by summing out all possible subdivision schemes.

$$p(\mathbf{c}) = \sum_{\mathcal{S}} p(\mathbf{c}|\mathcal{S})p(\mathcal{S}) \quad (2.16)$$

where $p(\mathcal{S})$ is the prior distribution of subdivision schemes. For example, one can select possible subdivision schemas as $\mathcal{S}_1 = [2, 2, 2]$, $\mathcal{S}_2 = [3, 2, 2]$, $\mathcal{S}_3 = [2, 3, 2]$. If we have a preference towards the time signature (4/4), the prior can be taken as $p(\mathcal{S}) = [1/2, 1/4, 1/4]$. In general, this choice should reflect the relative frequency of time signatures. We propose the following form for the prior of $\mathcal{S} = [s_i]$

Table 2.1: $w(s_i)$

s_i	2	3	5	7	11	13	17	o/w
$w(s_i)$	0	1	2	3	4	5	6	∞

$$p(\mathcal{S}) \propto e^{-\xi \sum_i w(s_i)} \quad (2.17)$$

where $w(s_i)$ is a simple weighting function given in Table 2.1. This form prefers subdivisions by small prime numbers, which reflects the intuition that rhythmic subdivisions by prime numbers such as 7 or 11 are far less common than subdivisions such as 2 or 3. The parameter ξ distributes probability mass over the primes. When $\xi = 0$, all subdivision schemata are equally probable. As $\xi \rightarrow \infty$, only subdivisions with $s_i = 2$ have non-zero probability.

2.3 Verification of the Model

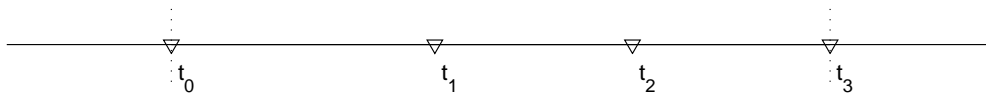
To choose the likelihood $p(\mathbf{t}|\mathbf{c})$ and the prior $p(\mathbf{c})$ in a way which is perceptually meaningful, we analyzed data obtained from an psychoacoustical experiment where ten well trained subjects (nine conservatory students and a conservatory professor) have participated (Desain, Aarts, Cemgil, Kappen, van Thienen, & Trilsbeek, 1999). The experiment consisted of a perception task and a production task.

2.3.1 Perception Task

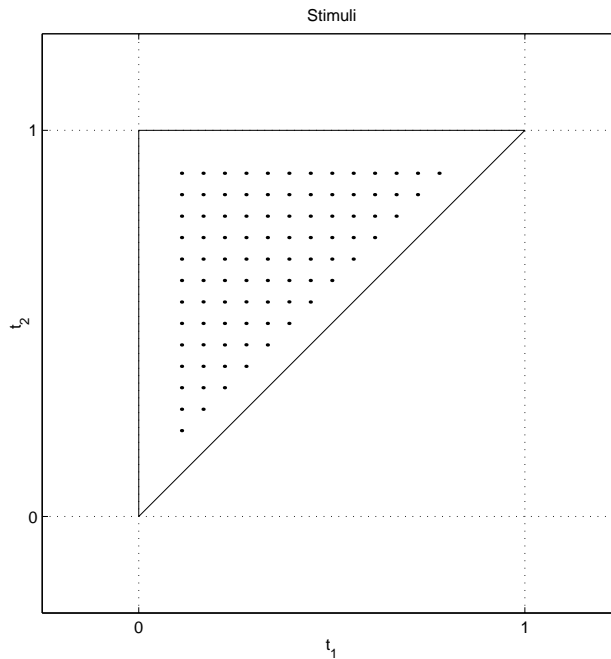
In the perception task the subjects were asked to transcribe 91 different *stimuli*. These rhythms consisted of four onsets $t_0 \dots t_3$ where t_0 and t_3 were fixed and occur exactly on the beat (Figure 2.9). First a beat is provided to subjects (count in), and then the stimulus is repeated 3 times with an empty bar between each repetition. Subjects were allowed to use any notation as a response and listen to the stimulus as much as they wanted. In total, subjects used 125 different notations, from which 57 were used only once and 42 are used more than three times. An example is depicted in Figure 2.10(a). From this data, we estimate the posterior as

$$q(\mathbf{c}_j | \mathbf{t}_k) = n_k(\mathbf{c}_j) / \sum_j n_k(\mathbf{c}_j)$$

where $n_k(\mathbf{c}_j)$ denotes the number of times the stimulus \mathbf{t}_k is associated with the notation \mathbf{c}_j .



(a) Stimulus



(b) Stimuli for the perception experiment. The dots denote the rhythms \mathbf{t}_k , where $k = 1 \dots 91$. Grid spacing is 56ms.

Figure 2.9: Stimulus of the Perception Task

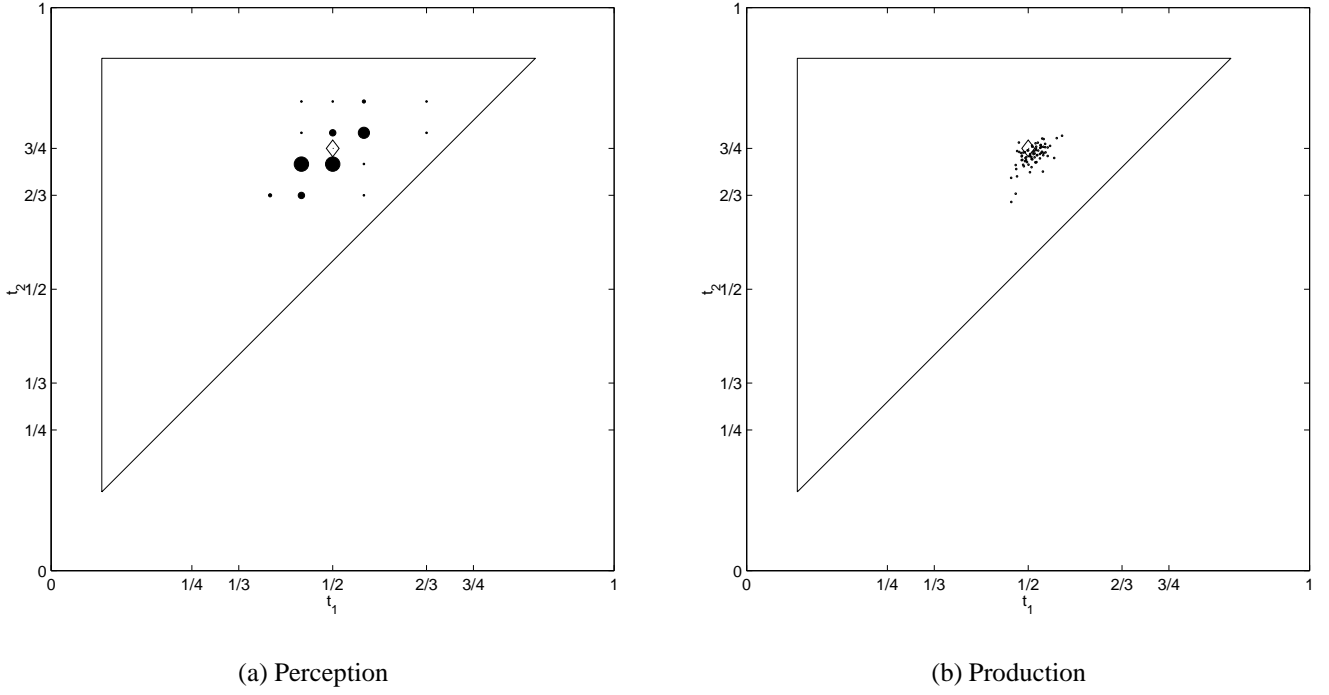


Figure 2.10: Perception and Production of the rhythm [2 1 1] ($\mathbf{c} = [0.5 \ 0.75]$). The diamond corresponds to the mechanical performance. In 2.10(a), the size of the circles is proportional to the estimated posterior $q(\mathbf{c}_j | \mathbf{t}_k)$. In 2.10(b), the dots correspond to performances of the rhythm.

2.3.2 Production Task

In the production task the subjects are asked to perform the rhythms that they have notated in the perception task. An example is shown in Figure 2.10(a). For each notation \mathbf{c}_j we assume a gaussian distribution where

$$\hat{q}(\mathbf{t} | \mathbf{c}_j) = \mathcal{N}(\mu_j, \Sigma_j) \quad (2.18)$$

The mean and the covariance matrix are estimated from production data by

$$\mu_j = \frac{1}{N_j} \sum_k \mathbf{t}_{k,j} \quad (2.19)$$

$$\Sigma_j = \frac{1}{N_j - 1} \sum_{k,l} (\mathbf{t}_{k,j} - \mu_j)(\mathbf{t}_{l,j} - \mu_j)^T \quad (2.20)$$

where $\mathbf{t}_{k,j}$ is the k 'th performance of \mathbf{c}_j and N_j is the total count of these performances in the data set. In Section 2.2.5 we proposed a model in which the correlation between two onset decreases with increasing inter-onset interval. The correlation coefficient and the estimated error bars are depicted in Figure 2.11, where we observe that the correlation decreases with increasing distance between onsets.

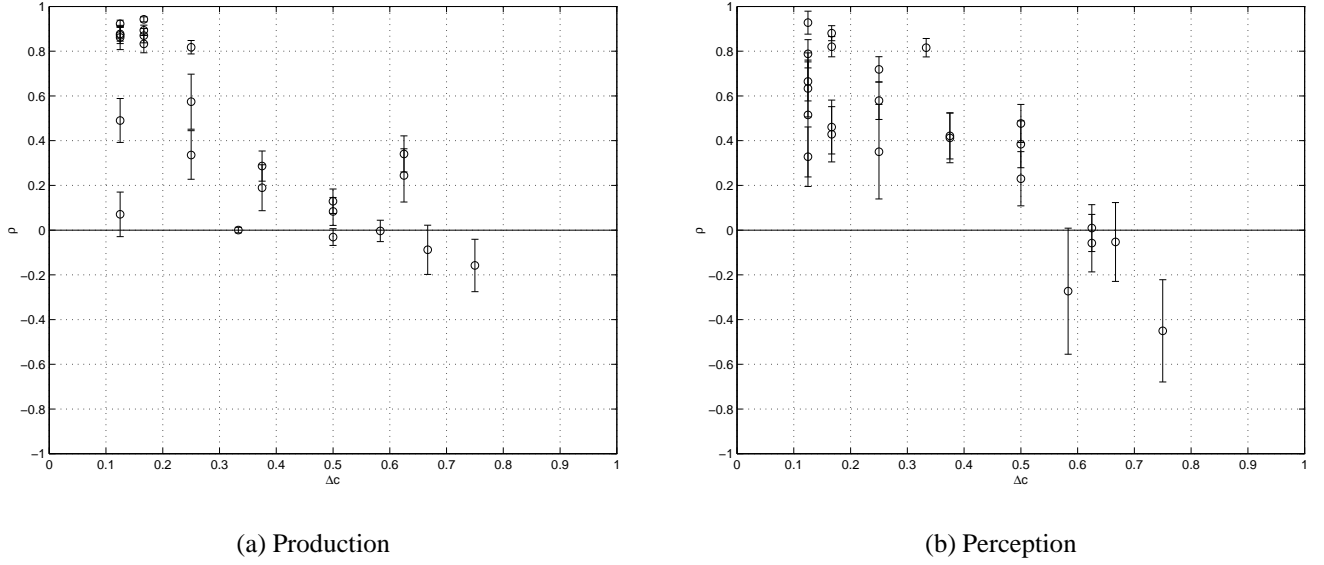


Figure 2.11: Estimated correlation coefficient as a function of $\Delta c = (c_2 - c_1)$ on all subject responses.

2.3.3 Estimation of model parameters

The probabilistic model $p(\mathbf{c}|\mathbf{t})$ described in the previous section can be fitted by minimizing the “distance” to the estimated target $q(\mathbf{c}|\mathbf{t})$. A well known distance measure between two probability distributions is the Kullback-Leiber divergence (Cover & Thomas, 1991) which is given as

$$\text{KL}(q||p) = \int d\mathbf{x} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \quad (2.21)$$

The integration is replaced by summation for discrete probability distributions. It can be shown (Cover & Thomas, 1991) that $\text{KL}(q||p) \geq 0$ for any q, p and vanishes if and only if $q = p$.

KL divergence can be interpreted as a “weighted average” of the function $\log \frac{q(\mathbf{x})}{p(\mathbf{x})}$ with respect to weighting function $q(\mathbf{x})$. If $q(\mathbf{x})$ and $p(\mathbf{x})$ are significantly different for some \mathbf{x} (for which $q(\mathbf{x})$ is sufficiently large), the KL divergence would be also large and would indicate that the distributions are different. On the other if the distributions have almost the same shape, $\frac{q(\mathbf{x})}{p(\mathbf{x})} \approx 1$ for all \mathbf{x} , and KL would be close to zero since $\log(1) = 0$.

The KL divergence is an appropriate measure for the rhythm quantization problem. We observe that for many stimuli, subjects give different responses and consequently it is difficult to choose just one “correct” notation for a particular stimulus. In other words, the target distribution $q(\mathbf{c}|\mathbf{t})$ has its mass distributed among several codevectors. By minimizing the KL divergence one can approximate the posterior distribution by preserving this intrinsic uncertainty.

The optimization problem for the perception task can be set as

$$\begin{aligned} \min . \quad & \text{KL}(q(\mathbf{c}|\mathbf{t})s(\mathbf{t})||p(\mathbf{c}|\mathbf{t})s(\mathbf{t})) \\ \text{s.t.} \quad & \sigma > 0 \\ & -1 < \eta < 1 \\ & \lambda, \xi, \gamma \text{ unconstrained} \end{aligned} \quad (2.22)$$

where $s(\mathbf{t}) \propto \sum_k \delta(\mathbf{t} - \mathbf{t}_k)$ is the distribution of the stimuli. This is a distribution, which has

positive mass only on the stimuli points \mathbf{t}_k . This measure forces the model to fit the estimated posterior at each stimulus point \mathbf{t}_k . We note that

$$p(\mathbf{c}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{c}; \sigma, \lambda, \eta)p(\mathbf{c}; \xi, \gamma)}{\sum_{\mathbf{c}} p(\mathbf{t}|\mathbf{c}; \sigma, \lambda, \eta)p(\mathbf{c}; \xi, \gamma)} \quad (2.23)$$

This is in general a rather difficult optimization problem due to the presence of the denominator. Nevertheless, since the model has only five free parameters, we were able to minimize Eq. 2.22 by a standard BFGS Quasi-Newton algorithm (MATLAB function `fminu`). In our simulations, we observed that the objective function is rather smooth and the optimum found is not sensitive to starting conditions, which suggests that there are not many local minima present.

2.3.4 Results

The model is trained on a subset of the perception data by minimizing Eq. 2.22. In the training, we used 112 different notations (out of 125 that the subjects used in total), which could be generated by one of the subdivision schemas in Table 2.2. To identify the relative importance of model parameters, we optimized Eq. 2.22 by clamping some parameters. We use a labeling of different models as follows: Model-I is the “complete” model, where all parameters are unclamped. Model-II is an onset quantizer ($\Sigma = \sigma^2 \mathbf{I}$), where only prior parameters are active. Model-III is (almost) an IOI quantizer where the correlation between onsets is taken to be $\rho = 0.98$. Model-IV is similar to Model I with the simplification that the covariance matrix is constant for all codevectors. Since $\lambda = 0$, $\rho = \eta$. Model-V is an onset quantizer with a flat prior, similar to the quantizers used in commercial notation packages and Model-VI has only the performance model parameters active.

In Model-VII, the parameters of the performance model $p(\mathbf{t}|\mathbf{c})$ are estimated from the production data. The model is fitted to the production data \hat{q} by minimizing

$$\text{KL}(\hat{q}(\mathbf{t}|\mathbf{c})q(\mathbf{c})||p(\mathbf{t}|\mathbf{c})q(\mathbf{c})) \quad (2.24)$$

where $q(\mathbf{c}_j) = \sum_k n_k(\mathbf{c}_j) / \sum_{k,j} n_k(\mathbf{c}_j)$, i.e. a histogram obtained by counting the subject responses in the perception experiment.

Although approximating the posterior at stimuli points is our objective in the optimization, for automatic transcription we are also interested into the classification performance. At each stimuli \mathbf{t}_k , if we select the response which the subjects have chosen the most, i.e. $\mathbf{c}_k^* = \arg \max_{\mathbf{c}} q(\mathbf{c}|\mathbf{t}_k)$, we can achieve maximum possible classification rate on this dataset, which is given as

$$\text{CR}_{\text{Target}} = \frac{n_k(\mathbf{c}_k^*)}{Z} \times 100 \quad (2.25)$$

Here, $Z = \sum_{k,\mathbf{c}} n_k(\mathbf{c}_k^*)$, the total number of measurements. Similarly, if we select the codevector with the highest predicted posterior $\mathbf{c}_k^* = \arg \max_{\mathbf{c}} p(\mathbf{c}|\mathbf{t}_k)$ at each stimulus, we achieve the classification rate of the Model denoted as CR_{Model} . The results are shown in Table 2.3. The clamped parameters are tagged with an ‘=’ sign. The results are for a codebook consisting of 112 codevectors, which the subjects have used in their responses and could have been generated by one of the subdivisions in Table 2.2.

Model-I performs the best in terms of the KL divergence, however the marginal benefit obtained by choosing a correlation structure, which decreases with increasing onset distances (obtained by varying λ) is rather small. One can achieve almost the same performance by having a constant correlation between onsets (Model-IV). By comparing Model-IV to Models II and III, we can say that under the given prior distribution the subjects are employing a quantization strategy, which is somehow between a pure onset quantization and IOI-quantization. The choice of the prior

i	\mathcal{S}_i
1	[2, 2, 2, 2]
2	[3, 2, 2]
3	[3, 3, 2]
4	[5, 2]
5	[7, 2]
6	[11]
7	[13]
8	[5, 3]
9	[17]
10	[7, 3]

Table 2.2: Subdivisions

Model	Prior		Likelihood			Results	
Label	ξ	γ	σ	λ	η	KL	$CR_{\text{Model}}/CR_{\text{Target}}$
I	1.35	0.75	0.083	2.57	0.66	1.30	77.1
II	1.34	0.75	0.086	= 0	= 0	1.41	71.3
III	1.33	0.77	0.409	= 0	= 0.98	1.96	51.4
IV	1.34	0.74	0.084	= 0	0.39	1.34	75.3
V	= 0	= 0	0.085	= 0	= 0	1.92	29.7
VI	= 0	= 0	0.083	2.54	0.66	1.89	32.7
VII	1.43	0.79	! 0.053	! 3.07	! 0.83	1.89	84.3

Table 2.3: Optimization Results. $CR_{\text{Target}} = 48.0$. Values tagged with a ‘=’ are fixed during optimization. Values estimated from the production experiment are tagged with a ‘!’. The meanings of the columns are explained in the text.

is very important which can be seen from the results of Model-V and Model-VI, which perform poor due to the flat prior assumption.

Model-VII suggests that for this data set (under the assumption that our model is correct) the perception and production processes are different. This is mainly due to the spread parameter σ , which is smaller for the production data. The interpretation of this behavior is that subjects deviate less from the mechanical mean in a performance situation. However, this might be due to the fact that performances were carried out in lack of any context, which forces the subjects to concentrate on exact timing. It is interesting to note that almost the same correlation structure is reserved in both experiments. This suggests that there is some relation between the production and perception process. The classification performance of Model-VII is surprisingly high; it predicts the winner accurately. However the prediction of the posterior is poor, which can be seen by the high KL divergence score.

For visualization of the results we employ an interpolation procedure to estimate the target posterior at other points than the stimuli (See Appendix 2.4). The rhythm space can be tiled into regions of rhythms, which are quantized to the same codevector. Estimated tiles from experimental data are depicted in Figure 2.12(a).

In practice, it is not feasible to identify explicitly a subset of all possible codevectors, which have non-zero prior probability. For example, the number of notations which can be generated by subdivisions in Table 2.2 is 886 whereas the subjects used only 112 of these as a response. This subset must be predicted by the model as well. A simple grid quantizer tries to approximate this

subset by assigning a constant prior probability to codevectors only up to a certain threshold depth. The proposed prior model can be contrasted to this schema in that it distributes the probability mass in a perceptually more realistic manner. To visualize this, we generated a codebook consisting of all 886 codevectors. The tilings generated by Model-I and Model-V for this codebook are depicted in Figure 2.12(b) and 2.12(c). To compare the tilings, we estimate the ratio

$$\text{Match} = \frac{A_{\text{match}}}{A_{\text{total}}} \times 100 \quad (2.26)$$

where A_{match} is the area where the model matches with the target and A_{total} is the total area of the triangle. Note that this is just a crude approximation to the classification performance under the assumption that all rhythms are equally probable. The results are shown in Table. 2.4.

	I	II	III	IV	V	VI	VII
Match	58.8	53.5	36.1	59.0	3.8	3.1	56.7

Table 2.4: Amount of match between tilings generated by the target and models

2.4 Discussion and Conclusion

In this article, we developed a vector quantizer for transcription of musical performances. We considered the problem in the framework of Bayesian statistics where we proposed a quantizer model. Experimentally, we observe that even for quantization of simple rhythms, well trained subjects give quite different answers, i.e. in many cases, there is not only one correct notation. In this respect, probabilistic modelling provides a natural framework.

The quantizer depends upon two probability models, a performance model and a prior. The performance model generalizes simple quantization strategies by taking the correlation structure in the music into account, for example onset quantization appears as a special case. The particular parametric form is shown to be perceptually meaningful and facilitates efficient implementation. It can also be interpreted as a suitable distance measure between rhythms.

The prior model can be interpreted as a complexity measure. In contrast to the likelihood, which has a rather standard form, the prior reflects our intuitive and subjective notion about the complexity of a notation and derives from consideration of time signatures and the hierarchical (i.e. tree-like) structure of musical rhythms.

The model is verified and optimized by data obtained from a psychoacoustical experiment. The optimization results suggest that prior and likelihood parameters can be optimized independently, since clamping one set of parameters affects the optimal values of others only very slightly. This property makes the interpretation of the model easier. Since we explicitly state the probability model, we can make comparisons between models by using the KL divergence as a goodness of fit measure. Indeed any other model which computes a posterior distribution $p(c|t)$ could be compared in a quantitative manner using this framework. A class of statistical tests to determine whether one model is significantly better than another is known as bootstrapping methods (Efron & Tibshirani, 1993). This methods can be used to estimate error bars on the KL measures to determine any significant difference between models.

We have to stress the point, that the particular parameter settings we find from data are not the ultimate way of doing quantization in every circumstance. First, the model is not using any other attributes of notes (e.g. duration, pitch), which may give additional information for better quantization. Second, we have not addressed the context information. Theoretically, such improvements

can be integrated by proposing more complex likelihood and prior models. As already demonstrated, since all the assumptions are stated as distributions, corresponding optimal parameters can be estimated from experimental data. A practical but important limitation is that parameter estimation in more complex models requires larger dataset otherwise the estimation can be subject to overfitting. A large dataset is difficult to collect since one effectively has to rely on psychoacoustical experiments, which are inherently limited in the number of experimental conditions one can impose (e.g. number of onsets, tempo, context e.t.c.). Nevertheless, we believe that the current framework is a consistent and principled way to investigate the quantization problem.

Acknowledgements

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs. The first author is thankful to David Barber for stimulating discussions.

Appendix 2.A Estimation of the posterior from subject responses

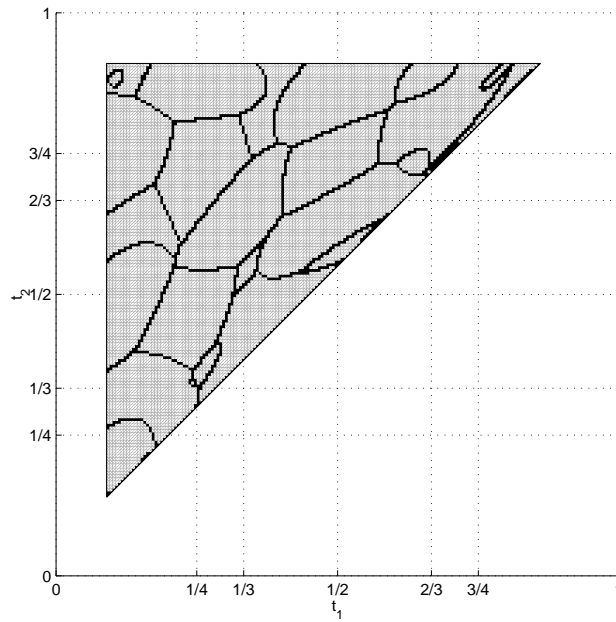
Let \mathbf{t}_k be the stimuli points. The histogram estimate at \mathbf{t}_k is denoted by $q(\mathbf{c}_j|\mathbf{t}_k)$. We define a kernel

$$G(\mathbf{t}; \mathbf{t}_0, \sigma) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{t}_0\|^2\right) \quad (2.27)$$

where $\|\mathbf{x}\|$ is the length of the vector \mathbf{x} . Then the posterior probability of \mathbf{c}_j at an arbitrary point \mathbf{t} is given as

$$q(\mathbf{c}_j|\mathbf{t}) = \sum_k \alpha_k(\mathbf{t})q(\mathbf{c}_j|\mathbf{t}_k) \quad (2.28)$$

where $\alpha_k(\mathbf{t}) = \frac{G(\mathbf{t};\mathbf{t}_k,\sigma)}{\sum_r G(\mathbf{t};\mathbf{t}_r,\sigma)}$. We have taken $\sigma = 0.04$.



(a) Target

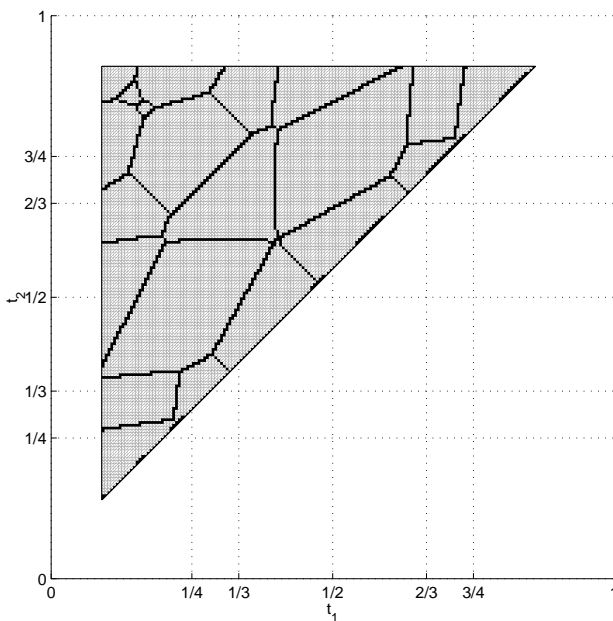
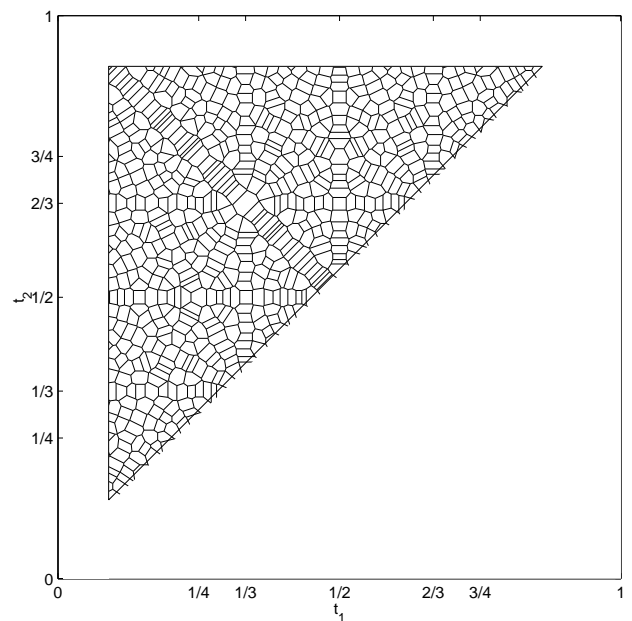
(b) Model-I: $(\xi, \gamma, \sigma, \lambda, \eta) = (1.35, 0.75, 0.083, 2.57, 0.66)$ (c) Model-V: $(\xi, \gamma, \sigma, \lambda, \eta) = (0, 0, 0.085, 0, 0)$

Figure 2.12: Tilings of the rhythm space by $\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmax}} p(\mathbf{c}|\mathbf{t})$. The tiles denote the sets of rhythms, which would be quantized to the same codevector. Both Model-I and Model-V use the same codebook of 886 codevectors. Since Model-V assigns the same prior probability to all codevectors, the best codevector is always the nearest codevector (in Euclidian distance) and consequently the rhythm space is highly fragmented.

Chapter 3

Tempo Tracking

We formulate tempo tracking in a Bayesian framework where a tempo tracker is modeled as a stochastic dynamical system. The tempo is modeled as a hidden state variable of the system and is estimated by a Kalman filter. The Kalman filter operates on a Tempogram, a wavelet-like multiscale expansion of a real performance. An important advantage of our approach is that it is possible to formulate both off-line or real-time algorithms. The simulation results on a systematically collected set of MIDI piano performances of Yesterday and Michelle by the Beatles shows accurate tracking of approximately 90% of the beats.

Adapted from: A. T. Cemgil, H. J. Kappen, P. Desain, and H. Honing. *On tempo tracking: Tempogram representation and Kalman filtering*. Journal of New Music Research, 28:4:259-273, 2001.

3.1 Introduction

An important and interesting subtask in automatic music transcription is tempo tracking: how to follow the tempo in a performance that contains expressive timing and tempo variations. When these tempo fluctuations are correctly identified it becomes much easier to separate the continuous expressive timing from the discrete note categories (i.e. quantization). The sense of tempo seems to be carried by the beats and thus tempo tracking is related to the study of beat induction, the perception of beats or pulse while listening to music (see (Desain & Honing, 1994)). However, it is still unclear what precisely constitutes tempo and how it relates to the perception of rhythmical structure. Tempo is a perceptual construct and cannot directly be measured in a performance.

In the context of tempo tracking, wavelet analysis and related techniques are already investigated by various researchers (Smith, 1999; Todd, 1994). A similar comb filter basis is used by (Scheirer, 1998). The tempogram is also related to the periodicity transform proposed by (Sethares & Staley, 1999), but uses a time localized basis. Kalman filters are already applied in the music domain such as polyphonic pitch tracking (Sterian, 1999) and audio restoration (Godsill & Rayner, 1998). From the modeling point of view, the framework discussed in this paper has also some resemblance to the work of (Sterian, 1999), who views transcription as a model based segmentation of a time-frequency image.

The outline of the paper is as follows: We first consider the problem of tapping along a “noisy” metronome and introduce the Kalman filter and its extensions. Subsequently, we introduce the Tempogram representation to extract beats from performances and discuss the probabilistic interpretation. Consequently, we discuss parameter estimation issues from data. Finally we report simulation results of the system on a systematically collected data set, solo piano performances of two Beatles songs, Yesterday and Michelle.

3.2 Dynamical Systems and the Kalman Filter

Mathematically, a dynamical system is characterized by a set of *state variables* and a set of *state transition equations* that describe how state variables evolve with time. For example, a perfect metronome can be described as a dynamical system with two state variables: a beat $\hat{\tau}$ and a period $\hat{\Delta}$. Given the values of state variables at $j - 1$ 'th step as $\hat{\tau}_{j-1}$ and $\hat{\Delta}_{j-1}$, the next beat occurs at $\hat{\tau}_j = \hat{\tau}_{j-1} + \hat{\Delta}_{j-1}$. The period of a perfect metronome is constant so $\hat{\Delta}_j = \hat{\Delta}_{j-1}$. By using vector notation and by letting $\mathbf{s}_j = [\hat{\tau}_j, \hat{\Delta}_j]^T$ we can write a linear state transition model as

$$\mathbf{s}_j = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{j-1} = \mathbf{A}\mathbf{s}_{j-1} \quad (3.1)$$

When the initial state $\mathbf{s}_0 = [\hat{\tau}_0, \hat{\Delta}_0]^T$ is given, the system is fully specified. For example if the metronom clicks at a tempo 60 beats per minute ($\hat{\Delta}_0 = 1$ sec.) and first click occurs at time $\hat{\tau}_0 = 0$ sec., next beats occur at $\hat{\tau}_1 = 1$, $\hat{\tau}_2 = 2$ e.t.c. Since the metronom is perfect the period stays constant.

Such a deterministic model is not realistic for natural music performance and can not be used for tracking the tempo in presence of tempo fluctuations and expressive timing deviations. Tempo fluctuations may be modeled by introducing a noise term that ‘‘corrupts’’ the state vector

$$\mathbf{s}_j = \mathbf{A}\mathbf{s}_{j-1} + \mathbf{v}_j \quad (3.2)$$

where \mathbf{v} is a Gaussian random vector with mean 0 and diagonal covariance matrix \mathbf{Q} , i.e. $\mathbf{v} \sim \mathcal{N}(0, \mathbf{Q})^1$. The tempo will drift from the initial tempo quickly if the variance of \mathbf{v} is large. On the other hand when $\mathbf{Q} \rightarrow 0$, we have the constant tempo case.

In a music performance, the actual beat $\hat{\tau}$ and the period $\hat{\Delta}$ can not be observed directly. By actual beat we refer to the beat interpretation that coincides with human perception when listening to music. For example, suppose, an expert drummer is tapping along a performance at the beat level and we assume her beats as the correct tempo track. If the task would be repeated on the same piece, we would observe each time a slightly different tempo track. As an alternative, suppose we would know the score of the performance and identify onsets that coincide with the beat. However, due to small scale expressive timing deviations, these onsets will be also noisy, i.e. we can at best observe ‘‘noisy’’ versions of actual beats. We will denote this noisy beat by τ in contrast to the actual but unobservable beat $\hat{\tau}$. Mathematically we have

$$\tau_j = \hat{\tau}_j + \mathbf{w}_j \quad (3.3)$$

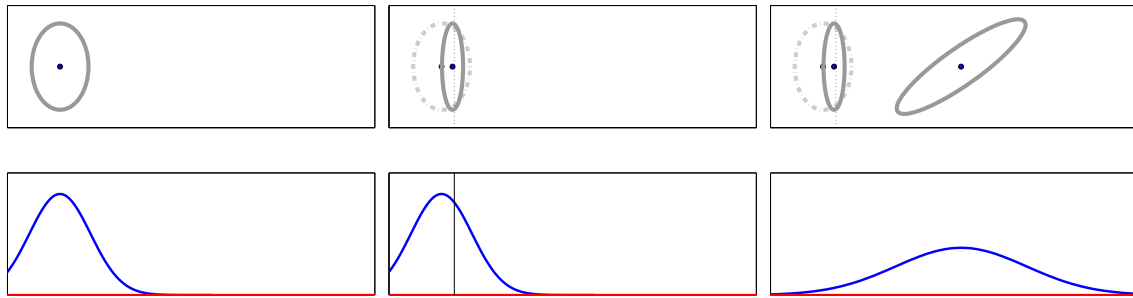
where $\mathbf{w}_j \sim \mathcal{N}(0, \mathbf{R})$. Here, τ_j is the beat at step j that we get from a (noisy) observation process. In this formulation, tempo tracking corresponds to the estimation of hidden variables $\hat{\tau}_j$ given observations upto j 'th step. We note that in a ‘‘blind’’ tempo tracking task, i.e. when the score is not known, the (noisy) beat τ_j can not be directly observed since there is no expert drummer who is tapping along, neither a score to guide us. The noisy-beat itself has to be *induced* from events in the music. In the next section we will present a technique to estimate both a noisy beat τ_j as well a noisy period Δ_j from a real performance.

¹A random vector \mathbf{x} is said to be Gaussian with mean μ and covariance matrix \mathbf{P} if it has the probability density

$$p(\mathbf{x}) = |2\pi\mathbf{P}|^{-1/2} \exp -\frac{1}{2}(\mathbf{x} - \mu)^T \mathbf{P}^{-1}(\mathbf{x} - \mu)$$

In this case we write $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{P})$

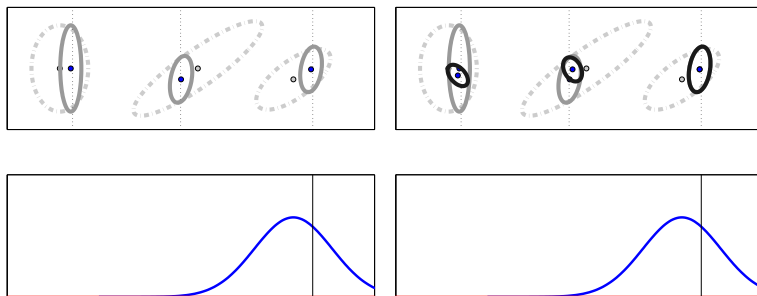
Equations 3.2 and 3.3 define a *linear dynamical system*, because all noises are assumed to be Gaussian and all relationships between variables are linear. Hence, all state vectors \mathbf{s}_j have Gaussian distributions. A Gaussian distribution is fully characterized by its mean and covariance matrix and in the context of linear dynamical systems, these quantities can be estimated very efficiently by a *Kalman filter* (Kalman, 1960; Roweis & Ghahramani, 1999). The operation of the filter is illustrated in Figure 3.1.



(a) The algorithm starts with the initial state estimate $\mathcal{N}(\mu_{1|0}, P_{1|0})$. In presence of no evidence this state estimate gives rise to a prediction in the observable τ space,

(b) The beat is observed at τ_1 , The state is updated to $\mathcal{N}(\mu_{1|1}, P_{1|1})$ according to the new evidence. Note that the uncertainty “shrinks”,

(c) On the basis of current state a new prediction $\mathcal{N}(\mu_{2|1}, P_{2|1})$ is made,



(d) Steps are repeated until all evidence is processed to obtain filtered estimates $\mathcal{N}(\mu_{j|j}, P_{j|j}), j = 1 \dots N$. In this case $N = 3$.

(e) Filtered estimates are updated by backtracking to obtain smoothed estimates $\mathcal{N}(\mu_{i|N}, P_{i|N})$ (Kalman smoothing).

Figure 3.1: Operation of the Kalman Filter and Smoother. The system is given by Equations 3.2 and 3.3. In each subfigure, the above coordinate system represents the hidden state space $[\hat{\tau}, \hat{\Delta}]^T$ and the below coordinate system represent the observable space τ . In the hidden space, the x and y axes represent the phase $\hat{\tau}$ period $\hat{\Delta}$ of the tracker. The ellipse and its center correspond to the covariance and the mean of the hidden state estimate $p(\mathbf{s}_j|\tau_1 \dots \tau_k) = \mathcal{N}(\mu_{j|k}, P_{j|k})$ where $\mu_{j|k}$ and $P_{j|k}$ denote the estimated mean and covariance given observations $\tau_1 \dots \tau_k$. In the observable space, the vertical axis represents the predictive probability distribution $p(\tau_j|\tau_{j-1} \dots \tau_1)$.

3.2.1 Extensions

The basic model can be extended in several directions. First, the linearity constraint on the Kalman filter can be relaxed. Indeed, in tempo tracking such an extension is necessary to ensure that the period $\hat{\Delta}$ is always positive. Therefore we define the state transition model in a warped space defined by the mapping $\omega = \log_2 \Delta$. This warping also ensures the perceptually more plausible assumption that tempo changes are relative rather than absolute. For example, under this warping, a deceleration from $\Delta \rightarrow 2\Delta$ has the same likelihood as an acceleration from $\Delta \rightarrow \Delta/2$.

The state space s_j can be extended with additional dynamic variables $\hat{\mathbf{a}}_j$. Such additional variables store information about the past states (e.g. in terms of acceleration e.t.c.) and introduce inertia to the system. Inertia reduces the random walk behavior in the state space and renders smooth state trajectories more likely. Moreover, this can result in more accurate predictions.

The observation noise \mathbf{w}_j can be modeled as a mixture of gaussians. This choice has the following rationale: To follow tempo fluctuations the observation noise variance \mathbf{R} should not be too “broad”. A broad noise covariance indicates that observations are not very reliable, so they have less effect to the state estimates. In the extreme case when $\mathbf{R} \rightarrow \infty$, all observations are practically missing so the observations have no effect on state estimates. On the other hand, a narrow \mathbf{R} makes the filter sensitive to outliers since the same noise covariance is used regardless of the distance of an observation from its prediction. Outliers can be explicitly modeled by using a mixture of Gaussians, for example one “narrow” Gaussian for normal operation, and one “broad” Gaussian for outliers. Such a switching mechanism can be implemented by using a discrete variable c_j which indicates whether the j 'th observation is an outlier or not. In other words we use a different noise covariance depending upon the value of c_j . Mathematically, we write this statement as $\mathbf{w}_j|c_j \sim \mathcal{N}(0, \mathbf{R}_c)$. Since c_j can not be observed, we define a prior probability $c_j \sim p(c)$ and sum over all possible settings of c_j , i.e. $p(\mathbf{w}_j) = \sum_{c_j} p(c_j)p(\mathbf{w}_j|c_j)$. In Figure 3.2 we compare a switching Kalman filter and a standard Kalman filter. A switch variable makes a system more robust against outliers and consequently more realistic state estimates can be obtained. For a review of more general classes of switching Kalman filters see (Murphy, 1998).

To summarize, the dynamical model of the tempo tracker is given by

$$\hat{\tau}_j = \hat{\tau}_{j-1} + 2^{\hat{\omega}_{j-1}} \quad (3.4)$$

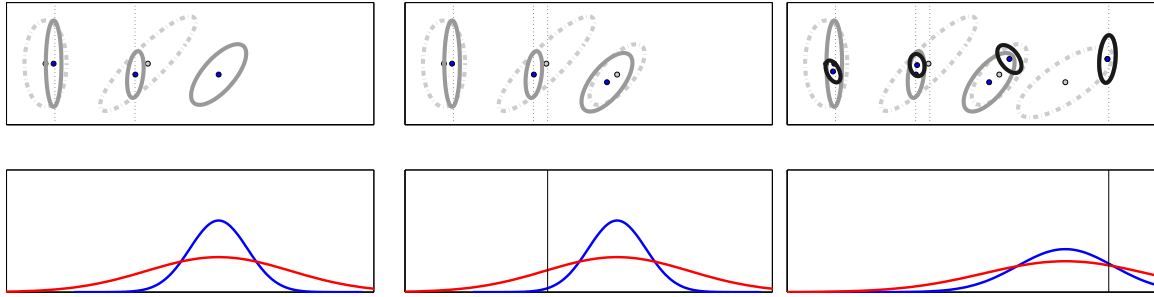
$$\begin{pmatrix} \hat{\omega}_j \\ \hat{\mathbf{a}}_j \end{pmatrix} = \mathbf{A} \begin{pmatrix} \hat{\omega}_{j-1} \\ \hat{\mathbf{a}}_{j-1} \end{pmatrix} + \mathbf{v}_j \quad (3.5)$$

$$\begin{pmatrix} \tau_j \\ \omega_j \end{pmatrix} = \begin{pmatrix} \hat{\tau}_j \\ \hat{\omega}_j \end{pmatrix} + \mathbf{w}_j \quad (3.6)$$

where $\mathbf{v}_j \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{w}_j|c_j \sim \mathcal{N}(0, \mathbf{R}_c)$ and $c_j \sim p(c_j)$. We take c_j as a binary discrete switch variable. Note that, in Eq. 3.6 the observable space is two dimensional (includes both τ and ω), in contrast to one dimensional observable τ in Figure 3.2.

3.3 Tempogram Representation

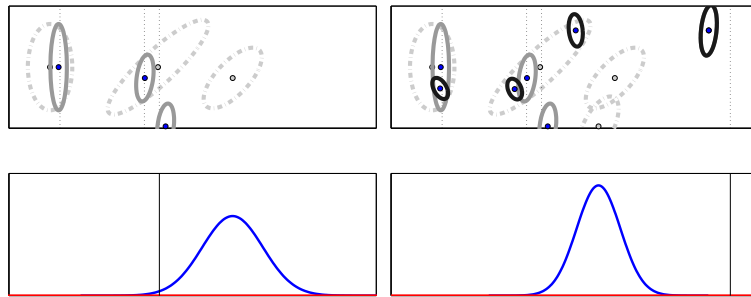
In the previous section, we have assumed that the beat τ_j is observed at each step j . In a real musical situation, however, the beat can not be observed directly from performance data. The sensation of a beat emerges from a *collection* of events rather than, say, single onsets. For example, a syncopated rhythm induces beats which do not necessarily coincide with an onset.



(a) Based on the state estimate $\mathcal{N}(\mu_{2|2}, P_{2|2})$ the next state is predicted as $\mathcal{N}(\mu_{3|2}, P_{3|2})$. When propagated through the measurement model, we obtain $p(\tau_3|\tau_2, \tau_1)$, which is a mixture of Gaussians where the mixing coefficients are given by $p(c)$,

(b) The observation τ_3 is way off the mean of the prediction, i.e. it is highly likely an outlier. Only the broad Gaussian is active, which reflects the fact that the observations are expected to be very noisy. Consequently, the updated state estimate $\mathcal{N}(\mu_{3|3}, P_{3|3})$ is not much different than its prediction $\mathcal{N}(\mu_{3|2}, P_{3|2})$. However, the uncertainty in the next prediction $\mathcal{N}(\mu_{4|3}, P_{4|3})$ will be higher,

(c) After all observations are obtained, the smoothed estimates $\mathcal{N}(\mu_{j|4}, P_{j|4})$ are obtained. The estimated state trajectory shows that the observation τ_3 is correctly interpreted as an outlier.



(d) In contrast to the switching Kalman filter, the ordinary Kalman filter is sensitive against outliers. In contrast to (b), the updated state estimate $\mathcal{N}(\mu_{3|3}, P_{3|3})$ is way off the prediction.

(e) Consequently a very “jumpy” state trajectory is estimated. This is simply due to the fact that the observation model does not account for presence of outliers.

Figure 3.2: Comparison of a standard Kalman filter with a switching Kalman filter.

In this section, we will define a probability distribution which assigns probability masses to all possible beat interpretations given a performance. The Bayesian formulation of this problem is

$$p(\tau, \omega | \mathbf{t}) \propto p(\mathbf{t} | \tau, \omega) p(\tau, \omega) \quad (3.7)$$

where \mathbf{t} is an onset list. In this context, a *beat interpretation* is the tuple τ (local beat) and ω (local log-period).

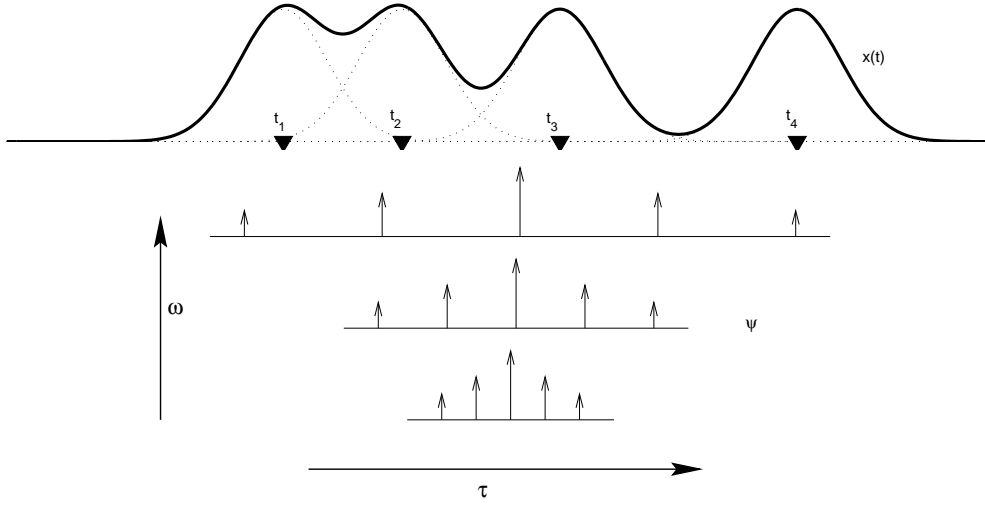


Figure 3.3: Tempogram Calculation. The continuous signal $x(t)$ is obtained from the onset list by convolution with a Gaussian function. Below, three different basis functions ψ are shown. All are localized at the same τ and different ω . The tempogram at (τ, ω) is calculated by taking the inner product of $x(t)$ and $\psi(t; \tau, \omega)$. Due to the sparse nature of the basis functions, the inner product operation can be implemented very efficiently.

The first term $p(\mathbf{t}|\tau, \omega)$ in Eq.3.7 is the probability of the onset list \mathbf{t} given the tempo track. Since \mathbf{t} is actually observed, $p(\mathbf{t}|\tau, \omega)$ is a function of τ and ω and is thus called the *likelihood* of τ and ω . The second term $p(\tau, \omega)$ in Eq.3.7 is the *prior* distribution. The prior can be viewed as a function which weights the likelihood on the (τ, ω) space. It is reasonable to assume that the likelihood $p(\mathbf{t}|\tau, \omega)$ is high when onsets $[t_i]$ in the performance coincide with the beats of the tempo track. To construct a likelihood function having this property we propose a similarity measure between the performance and a *local* constant tempo track. First we define a continuous time signal $x(t) = \sum_{i=1}^I G(t - t_i)$ where we take $G(t) = \exp(-t^2/2\sigma_x^2)$, a Gaussian function with variance σ_x^2 . We represent a local tempo track as a pulse train $\psi(t; \tau, \omega) = \sum_{m=-\infty}^{\infty} \alpha_m \delta(t - \tau - m2^\omega)$ where $\delta(t - t_0)$ is a Dirac delta function, which represents an impulse located at t_0 . The coefficients α_m are positive constants such that $\sum_m \alpha_m$ is a constant. (See Figure 3.3). In real-time applications, where causal analysis is desirable, α_m can be set to zero for $m > 0$. When α_m is a sequence of form $\alpha_m = \alpha^m$, where $0 < \alpha < 1$, one has the infinite impulse response (IIR) comb filters used by (Scheirer, 1998) which we adopt here. We define the *tempogram* of $x(t)$ at each (τ, ω) as the inner product

$$\text{Tg}_x(\tau, \omega) = \int dt x(t)\psi(t; \tau, \omega) \quad (3.8)$$

The tempogram representation can be interpreted as the response of a comb filter bank and is analogous to a multiscale representation (e.g. the wavelet transform), where τ and ω correspond to transition and scaling parameters (Rioul & Vetterli, 1991; Kronland-Martinet, 1988).

The tempogram parameters have simple interpretations. The filter coefficient α adjust the time locality of basis functions. When $\alpha \rightarrow 1$, basis functions ψ extend to infinity and locality is lost. For $\alpha \rightarrow 0$ the basis degenerates to a single Dirac pulse and the tempogram is effectively equal to $x(t)$ for all ω and thus gives no information about the local period.

The variance parameter σ_x corresponds to the amount of small scale expressive deviation in an onsets timing. If σ_x would be large, the tempogram gets “smeared-out” and all beat interpretations

become almost equally likely. When $\sigma_x \rightarrow 0$, we get a very “spiky” tempogram, where most beat interpretations have zero probability.

In Figure 3.4 we show a tempogram obtained from a simple onset sequence. We define the likelihood as $p(\mathbf{t}|\tau, \omega) \propto \exp(\mathbf{T}g_x(\tau, \omega))$. When combined with the prior, the tempogram gives an estimate of likely beat interpretations (τ, ω) .

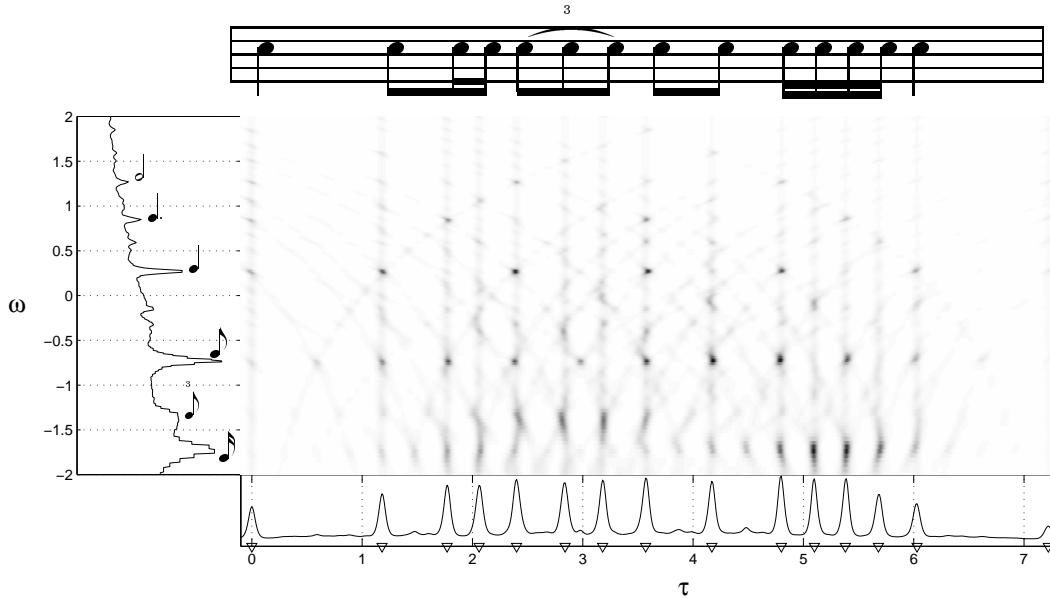


Figure 3.4: A simple rhythm and its Tempogram. x and y axes correspond to τ and ω respectively. The bottom figure shows the onset sequence (triangles). Assuming flat priors on τ and ω , the curve along the ω axis is the marginal $p(\omega|\mathbf{t}) \propto \int d\tau \exp(\mathbf{T}g_x(\tau, \omega))$. We note that $p(\omega|\mathbf{t})$ has peaks at ω , which correspond to quarter, eighth and sixteenth note level as well as dotted quarter and half note levels of the original notation. This distribution can be used to estimate a reasonable initial state.

3.4 Model Training

In this section, we review the techniques for parameter estimation. First, we summarize the relationships among variables by using a *graphical model*. A graphical model is a directed acyclic graph, where nodes represent variables and missing directed links represent conditional independence relations. The distributions that we have specified so far are summarized in Table 3.1.

Model	Distribution	Parameters
State Transition (Eq. 3.5)	$p(\mathbf{s}_{j+1} \mathbf{s}_j)$	\mathbf{A}, \mathbf{Q}
(Switching) Observation (Eq. 3.6)	$p(\tau_j, \omega_j \mathbf{s}_j, c_j)$	\mathbf{R}_c
Switch prior (Eq. 3.6)	$p(c_j)$	p_c
Tempogram (Eq.3.8)	$p(\mathbf{t} \tau_j, \omega_j)$	σ_x, α

Table 3.1: Summary of conditional distributions and their parameters.

The resulting graphical model is shown in Figure 3.5. For example, the graphical model has a directed link from \mathbf{s}_j to \mathbf{s}_{j+1} to encode $p(\mathbf{s}_{j+1}|\mathbf{s}_j)$. Other links towards \mathbf{s}_{j+1} are missing.

In principle, we could jointly optimize all model parameters. However, such an approach would be computationally very intensive. Instead, at the expense of getting a suboptimal solution, we

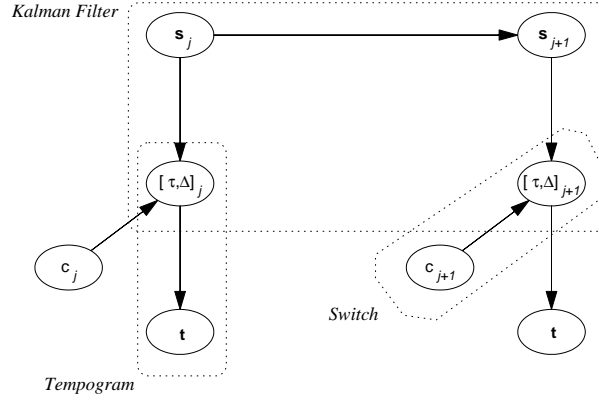


Figure 3.5: The Graphical Model

will assume that we observe the noisy tempo track τ_j . This observation effectively “decouples” the model into two parts (See Fig. 3.5), (i) The Kalman Filter (State transition model and Observation (Switch) model) and (ii) Tempogram. We will train each part separately.

3.4.1 Estimation of τ_j from performance data

In our studies, a score is always available, so we extract τ_j from a performance \mathbf{t} by matching the notes that coincide with the beat (quarter note) level and the bar (whole note). If there are more than one note on a beat, we take the median of the onset times.² For each performance, we compute $\omega_j = \log_2(\tau_{j+1} - \tau_j)$ from the extracted noisy beats $[\tau_j]$. We denote the resulting tempo track $\{\tau_1, \omega_1 \dots \tau_j, \omega_j \dots \tau_J, \omega_J\}$ as $\{\tau_{1:J}, \omega_{1:J}\}$.

3.4.2 Estimation of state transition parameters

We estimate the state transition model parameters \mathbf{A} and \mathbf{Q} by an EM algorithm (Ghahramani & Hinton, 1996) which learns a linear dynamics in the ω space. The EM algorithm monotonically increases $p(\{\tau_{1:J}, \omega_{1:J}\})$, i.e. the likelihood of the observed tempo track. Put another way, the parameters \mathbf{A} and \mathbf{Q} are adjusted in such a way that, at each j , the probability of the observation is maximized under the predictive distribution $p(\tau_j, \omega_j | \tau_{j-1}, \omega_{j-1}, \dots, \tau_1, \omega_1)$. The likelihood is simply the height of the predictive distribution evaluated at the observation (See Figure 3.1).

3.4.3 Estimation of switch parameters

The observation model is a Gaussian mixture with diagonal \mathbf{R}_c and prior probability p_c . We could estimate \mathbf{R}_c and p_c jointly with the state transition parameters \mathbf{A} and \mathbf{Q} . However, then the noise model would be totally independent from the tempogram representation. Instead, the observation noise model should reflect the uncertainty in the tempogram; for example the expected amount of deviations in (τ, ω) estimates due to spurious local maxima. To estimate the “tempogram noise” by standard EM methods, we sample from the tempogram around each $[\hat{\tau}_j, \hat{\omega}_j]$, i.e. we sample τ_j and ω_j from the posterior distribution $p(\tau_j, \omega_j | \hat{\tau}_j, \hat{\omega}_j, \mathbf{t}; \mathbf{Q}) \propto p(\mathbf{t} | \tau_j, \omega_j) p(\tau_j, \omega_j | \hat{\tau}_j, \hat{\omega}_j; \mathbf{Q})$. Note

²The scores do not have notes on each beat. We interpolate missing beats by using a switching Kalman filter with parameters $\mathbf{Q} = \text{diag}([0.01^2, 0.05^2])$, $\mathbf{R}_1 = 0.01^2$, $\mathbf{R}_2 = 0.3^2$, $\mathbf{A} = 1$ and $p(c) = [0.999, 0.001]$.

that $[\hat{\tau}_j, \hat{\omega}_j]$ are estimated during the E step of the EM algorithm when finding the parameters \mathbf{A} and \mathbf{Q} .

3.4.4 Estimation of Tempogram parameters

We have already defined the tempogram as a likelihood $p(\mathbf{t}|\tau, \omega; \theta)$ where θ denotes the tempogram parameters (e.g. $\theta = \{\alpha, \sigma_x\}$). If we assume a uniform prior $p(\tau, \omega)$ then the posterior probability can be written as

$$p(\tau, \omega|\mathbf{t}; \theta) = \frac{p(\mathbf{t}|\tau, \omega; \theta)}{p(\mathbf{t}|\theta)} \quad (3.9)$$

where the normalization constant is given by $p(\mathbf{t}|\theta) = \int d\tau d\omega p(\mathbf{t}|\tau, \omega; \theta)$. Now, we can estimate tempogram parameters θ by a maximum likelihood approach. We write the log-likelihood of an observed tempo track $\{\tau_{1:J}, \omega_{1:J}\}$ as

$$\log p(\{\tau_{1:J}, \omega_{1:J}\}|\mathbf{t}; \theta) = \sum_j \log p(\tau_j, \omega_j|\mathbf{t}; \theta) \quad (3.10)$$

Note that the quantity in Equation 3.10 is a function of the parameters θ . If we have k tempo tracks in the dataset, the complete data log-likelihood is simply the sum of all individual log-likelihoods. i.e.

$$\mathcal{L} = \sum_k \log p(\{\tau_{1:J}, \omega_{1:J}\}^k|\mathbf{t}^k; \alpha, \sigma_x) \quad (3.11)$$

where \mathbf{t}^k is the k 'th performance and $\{\tau_{1:J}, \omega_{1:J}\}^k$ is the corresponding tempo track.

3.5 Evaluation

Many tempo trackers described in the introduction are often tested with ad hoc examples. However, to validate tempo tracking models, more systematic data and rigorous testing is necessary. A tempo tracker can be evaluated by systematically modulating the tempo of the data, for instance by applying instantaneous or gradual tempo changes and comparing the models responses to human behavior (Michon, 1967; Dannenberg, 1993). Another approach is to evaluate tempo trackers on a systematically collected set of natural data, monitoring piano performances in which the use of expressive tempo change is free. This type of data has the advantage of reflecting the type of data one expects automated music transcription systems to deal with. The latter approach was adopted in this study.

3.5.1 Data

For the experiment 12 pianists were invited to play arrangements of two Beatles songs, Michelle and Yesterday. Both pieces have a relatively simple rhythmic structure with ample opportunity to add expressiveness by fluctuating the tempo. The subjects consisted of four professional jazz players (PJ), four professional classical performers (PC) and four amateur classical pianists (AC). Each arrangement had to be played in three tempo conditions, three repetitions per tempo condition. The tempo conditions were normal, slow and fast tempo (all in a musically realistic range and all according to the judgment of the performer). We present here the results for twelve subjects (12 subjects \times 3 tempi \times 3 repetitions \times 2 pieces = 216 performances). The performances were recorded

on a Yamaha Disklavier Pro MIDI grand piano using Opcode Vision. To be able to derive tempo measurements related to the musical structure (e.g., beat, bar) the performances were matched with the MIDI scores using the structure matcher of (Heijink, Desain, & Honing, 2000) available in POCO (Honing, 1990). This MIDI data, as well as related software will be made available at URL's <http://www.mbfys.kun.nl/~cemgil> and <http://www.nici.kun.nl/mmm> (under the heading Download).

3.5.2 Kalman Filter Training results

We use the performances of Michelle as the training set and Yesterday as the test set. To find the appropriate filter order (Dimensionality of s) we trained Kalman filters of several orders on two rhythmic levels: the beat (quarter note) level and the bar (whole note) level. Figure 3.6 shows the training and testing results as a function of filter order.

Extending the filter order, i.e. increasing the the size of the state space loosely corresponds looking more into the past. At bar level, using higher order filters merely results in overfitting as indicated by decreasing test likelihood. In contrast, on the beat level, the likelihood on the test set also increases and has a jump around order of 7. Effectively, this order corresponds to a memory which can store state information from the past two bars. In other words, tempo fluctuations at beat level have some structure that a higher dimensional state transition model can make use of to produce more accurate predictions.

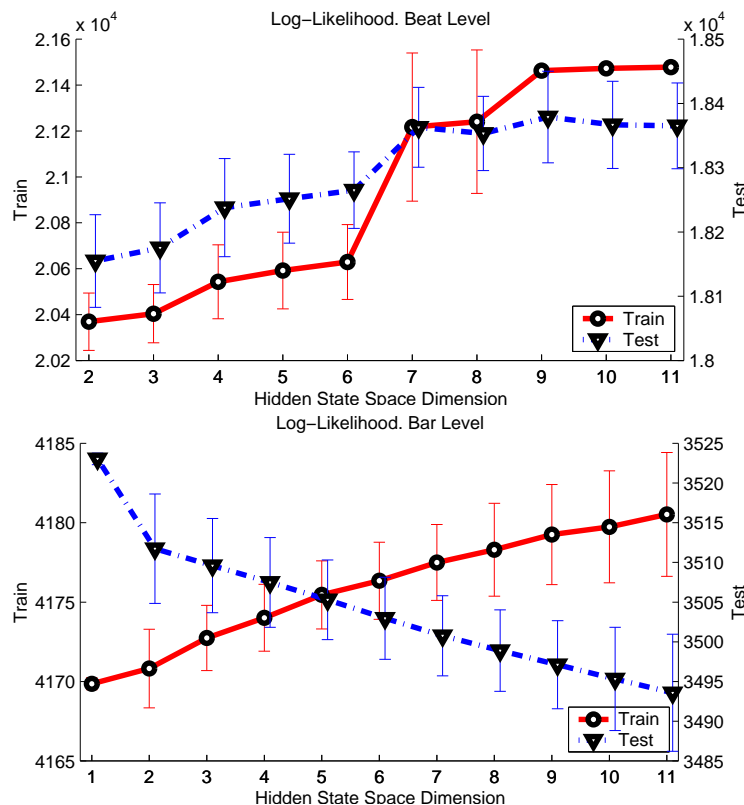


Figure 3.6: Kalman Filter training. Training Set: Michelle, Test Set: Yesterday.

3.5.3 Tempogram Training Results

We use a tempogram model with a first order IIR comb basis. This choice leaves two free parameters that need to be estimated from data, namely α , the coefficient of the comb filter and σ_x , the width of the Gaussian window. We obtain optimal parameter values by maximization of the log-likelihood in Equation 3.11 on the Michelle dataset. The optimal parameters are shown in Table 3.2.

	α	σ_x
Non-Causal	0.55	0.017
Causal	0.73	0.023

Table 3.2: Optimal tempogram parameters.

3.5.4 Initialization

To have a fully automated tempo tracker, the initial state s_0 has to be estimated from data as well. In the tracking experiments, we have initialized the filter to the beat level by computing a tempogram for the first 5 seconds of each performance. By assuming a flat prior on τ and ω we compute the posterior marginal $p(\omega|\mathbf{t}) = \int d\tau p(\omega, \tau|\mathbf{t})$. Note that this operation is just equivalent to summation along the τ dimension of the tempogram (See Figure 3.4). For the Beatles dataset, we have observed that for all performances of a given piece, the most likely log-period $\omega^* = \arg \max_{\omega} p(\omega|\mathbf{t})$ corresponds always to the same level, i.e. the ω^* estimate was always consistent. For “Michelle”, this level is the beat level and for “Yesterday” the half-beat (eighth note) level. The latter piece begins with an arpeggio of eight notes; based on onset information only, and without any other prior knowledge, half-beat level is also a reasonable solution. For “Yesterday”, to test the tracking performance, we corrected the estimate to the beat level.

We could estimate τ^* using a similar procedure, however since all performances in our data set started “on the beat”, we have chosen $\tau^* = t_1$, the first onset of the piece. All the other state variables $\hat{\mathbf{a}}_0$ are set to zero. We have chosen a broad initial state covariance $P_0 = 9\mathbf{Q}$.

3.5.5 Evaluation of tempo tracking performance

We evaluated the accuracy of the tempo tracking performance of the complete model. The accuracy of tempo tracking is measured by using the following criterion:

$$\rho(\psi, \mathbf{t}) = \frac{\sum_i \max_j W(\psi_i - t_j)}{(I + J)/2} \times 100$$

where $[\psi_i]$ $i = 1 \dots I$ is the target (true) tempo track and $[t_j]$ $j = 1 \dots J$ is the estimated tempo track. W is a window function. In the following results we have used a Gaussian window function $W(d) = \exp(-d^2/2\sigma_e^2)$. The width of the window is chosen as $\sigma_e = 0.04$ sec which corresponds roughly to the spread of onsets from their mechanical means during performance of short rhythms (Cemgil, Desain, & Kappen, 2000).

It can be checked that $0 \leq \rho \leq 100$ and $\rho = 100$ if and only if $\psi = \mathbf{t}$. Intuitively, this measure is similar to a normalized inner-product (as in the tempogram calculation); the difference is in the max operator which merely avoids double counting. For example, if the target is $\psi = [0, 1, 2]$ and we have $\mathbf{t} = [0, 0, 0]$, the ordinary inner product would still give $\rho = 100$ while only one beat is correct ($t = 0$). The proposed measure gives $\rho = 33$ in this case. The tracking index ρ can be

roughly interpreted as percentage of “correct” beats. For example, $\rho = 90$ effectively means that about 90 percent of estimated beats are in the near vicinity of their targets.

3.5.6 Results

To test the relative relevance of model components, we designed an experiment where we evaluate the tempo tracking performance under different conditions. We have varied the filter order and enabled or disabled switching. For this purpose, we trained two filters, one with a large (10) and one with a small (2) state space dimension on beat level (using the Michelle dataset). We have tested each model with both causal and non-causal tempograms. To test whether a tempogram is at all necessary, we propose a simple onset-only measurement model. In this alternative model, the next observation is taken as the nearest onset to the Kalman filter prediction. In case there are no onsets in 1σ interval of the prediction, we declare the observation as missing (Note that this is an implicit switching mechanism).

In Table 3.3 we show the tracking results averaged over all performances in the Yesterday dataset. The estimated tempo tracks are obtained by using a non-causal tempogram and Kalman filtering. In this case, Kalman smoothed estimates are not significantly different. The results suggest, that for the Yesterday dataset, a higher order filter or a (binary) switching mechanism does not improve the tracking performance. However, presence of a tempogram makes the tracking performance both more accurate and consistent (note the lower standard deviations). As a “base line” performance criteria, we also compute the best constant tempo track (by a linear regression to estimated tempo tracks). In this case, the average tracking index obtained from a constant tempo approximation is rather poor ($\rho = 28 \pm 18$), confirming that there is indeed a need for tempo tracking.

Filter order	Switching	tempogram	no tempogram
10	+	92 ± 7	75 ± 21
2	+	91 ± 9	75 ± 21
10	-	91 ± 6	73 ± 21
2	-	90 ± 9	73 ± 22

Table 3.3: Average tracking performance ρ and standard deviations on Yesterday dataset using a non-causal tempogram. + denotes the case when we have the switch prior $p(c) = [0.8, 0.2]$. – denotes the absence of a switching, i.e. the case when $p(c) = [1, 0]$.

We have repeated the same experiment with a causal tempogram and computed the tracking performance for predicted, filtered and smoothed estimates. In Table 3.4 we show the results for a switching Kalman filter. The results without switching are not significantly different. As one would expect, the tracking index with predicted estimates is lower. In contrast to a non-causal tempogram, smoothing improves the tempo tracking and results in a comparable performance as a non-causal tempogram.

Naturally, the performance of the tracker depends on the amount of tempo variations introduced by the performer. For example, the tempo tracker fails consistently for a subject who tends to use quite some tempo variation³.

We find that the tempo tracking performance is not significantly different among different groups (Table 3.5). However, when we consider the predictions, we see that the performances of professional classical pianists are less predictable. For different tempo conditions (Table 3.6) the results are also similar. As one would expect, for slower performances, the predictions are less

³This subject claimed to have never heard the Beatles songs before.

Filter order	causal		
	predicted	filtered	smoothed
10	74 ± 12	86 ± 9	91 ± 8
2	73 ± 12	85 ± 8	90 ± 8

Table 3.4: Average tracking performance ρ on Yesterday dataset. Figures indicate tracking index ρ followed by the standard deviation. The label “non-causal” refers to a tempogram calculated using non-causal comb filters. The labels predicted, filtered and smoothed refer to state estimates obtained by the Kalman filter/smoother.

accurate. This might have two potential reasons. First, the performance criteria ρ is independent of the absolute tempo, i.e. the window W is always fixed. Second, for slower performances there is more room for adding expression.

Subject Group	non-causal	causal			Best const.
	filtered	predicted	filtered	smoothed	
Prof. Jazz	95 ± 3	81 ± 7	92 ± 4	94 ± 3	34 ± 22
Amateur Classical	92 ± 8	74 ± 7	88 ± 5	92 ± 4	24 ± 19
Prof. Classical	89 ± 7	66 ± 14	82 ± 11	86 ± 11	27 ± 12

Table 3.5: Tracking Averages on subject groups. As a reference, the right most column shows the results obtained by the best constant tempo track. The label “non-causal” refers to a tempogram calculated using non-causal comb filters. The labels predicted, filtered and smoothed refer to state estimates obtained by the Kalman filter/smoother.

Condition	non-causal	causal			Best const.
	filtered	predicted	filtered	smoothed	
fast	94 ± 5	79 ± 9	90 ± 6	93 ± 6	39 ± 21
normal	92 ± 8	74 ± 9	88 ± 6	92 ± 4	25 ± 13
slow	90 ± 7	68 ± 14	84 ± 10	87 ± 11	21 ± 14

Table 3.6: Tracking Averages on tempo conditions. As a reference, the right most column shows the results obtained by the best constant tempo track. The label “non-causal” refers to a tempogram calculated using non-causal comb filters. The labels predicted, filtered and smoothed refer to state estimates obtained by the Kalman filter/smoother.

3.6 Discussion and Conclusions

In this paper, we have formulated a tempo tracking model in a probabilistic framework. The proposed model consist of a dynamical system (a Kalman Filter) and a measurement model (Tempogram). Although many of the methods proposed in the literature can be viewed as particular choices of a dynamical model and a measurement model, a Bayesian formulation exhibits several advantages in contrast to other models for tempo tracking. First, components in our model have natural probabilistic interpretations. An important and very practical consequence of such an interpretation is that uncertainties can be easily quantified and integrated into the system. Moreover, all desired quantities can be inferred consistently. For example once we quantify the distribution of

tempo deviations and expressive timing, the actual behavior of the tempo tracker arises automatically from these a-priori assumptions. This is in contrast to other models where one has to invent ad-hoc methods to avoid undesired or unexpected behavior on real data.

Additionally, prior knowledge (such as smoothness constraints in the state transition model and the particular choice of measurement model) are explicit and can be changed when needed. For example, the same state transition model can be used for both audio and MIDI; only the measurement model needs to be elaborated. Another advantage is that, for a large class of related models efficient inference and learning algorithms are well understood (Ghahramani & Hinton, 1996). This is appealing since we can train tempo trackers with different properties automatically from data. Indeed, we have demonstrated that all model parameters can be estimated from experimental data.

We have investigated several potential directions in which the basic dynamical model can be improved or simplified. We have tested the relative relevance of the filter order, switching and the tempogram representation on a systematically collected set of natural data. The dataset consists of polyphonic piano performances of two Beatles songs (Yesterday and Michelle) and contains a lot of tempo fluctuation as indicated by the poor constant tempo fits.

The test results on the Beatles dataset suggest that using a high order filter does not improve tempo tracking performance. Although beat level filters capture some structure in tempo deviations (and hence can generate more accurate predictions), this additional precision seems to be not very important in tempo tracking. This indifference may be due to the fact that training criteria (maximum likelihood) and testing criteria (tracking index), whilst related, are not identical. However, one can imagine scenarios where accurate prediction is crucial. An example would be a real-time accompaniment situation, where the application needs to generate events for the next bar.

Test results also indicate that a simple switching mechanism is not very useful. It seems that a tempogram already gives a robust local estimate of likely beat and tempo values so the correct beat can unambiguously be identified. The indifference of switching could as well be an artifact of the dataset which lacks extensive syncopations. Nevertheless, the switching noise model can further be elaborated to replace the tempogram by a rhythm quantizer (Cemgil et al., 2000).

To test the relevance of the proposed tempogram representation on tracking performance we have compared it to a simpler, onset based alternative. The results indicate that in the onset-only case, tracking performance significantly decreases, suggesting that a tempogram is an important component of the system.

It must be noted that the choice of a comb basis set for tempogram calculation is rather arbitrary. In principle, one could formulate a “richer” tempogram model, for example by including parameters that control the shape of basis functions. The parameters of such a model can similarly be optimized by likelihood maximization on target tempo tracks. Unfortunately, such an optimization (e.g. with a generic technique such as gradient descent) requires the computation of a tempogram at each step and is thus computationally quite expensive. Moreover, a model with many adjustable parameters might eventually overfit.

We have also demonstrated that the model can be used both online (filtering) and offline (smoothing). Online processing is necessary for real time applications such as automatic accompaniment and offline processing is desirable for transcription applications.

Chapter 4

Integrating Tempo Tracking and Quantization

We present a probabilistic generative model for timing deviations in expressive music performance. The structure of the proposed model is equivalent to a switching state space model. The switch variables correspond to discrete note locations as in a musical score. The continuous hidden variables denote the tempo. We formulate two well known music recognition problems, namely tempo tracking and automatic transcription (rhythm quantization) as filtering and maximum a posteriori (MAP) state estimation tasks. Exact computation of posterior features such as the MAP state is intractable in this model class, so we introduce Monte Carlo methods for integration and optimization. We compare Markov Chain Monte Carlo (MCMC) methods (such as Gibbs sampling, simulated annealing and iterative improvement) and sequential Monte Carlo methods (particle filters). Our simulation results suggest better results with sequential methods. The methods can be applied in both online and batch scenarios such as tempo tracking and transcription and are thus potentially useful in a number of music applications such as adaptive automatic accompaniment, score typesetting and music information retrieval.

Adapted from: A. T. Cemgil and H. J. Kappen. *Monte Carlo methods for tempo tracking and rhythm quantization*. *Journal of Artificial Intelligence Research*, 18:45-81, 2003.

4.1 Introduction

Automatic music transcription refers to extraction of a human readable and interpretable description from a recording of a musical performance. Traditional music notation is such a description that lists the pitch levels (notes) and corresponding timestamps.

Ideally, one would like to recover a score directly from the audio signal. Such a representation of the surface structure of music would be very useful in music information retrieval (Music-IR) and content description of musical material in large audio databases. However, when operating on sampled audio data from polyphonic acoustical signals, extraction of a score-like description is a very challenging auditory scene analysis task (Vercoe, Gardner, & Scheirer, 1998).

In this paper, we focus on a subproblem in music-ir, where we assume that exact timing information of notes is available, for example as a stream of MIDI¹ events from a digital keyboard.

¹Musical Instruments Digital Interface. A standard communication protocol especially designed for digital instruments such as keyboards. Each time a key is pressed, a MIDI keyboard generates a short message containing pitch and key velocity. A computer can tag each received message by a timestamp for real-time processing and/or recording into a file.

A model for tempo tracking and transcription from a MIDI-like music representation is useful in a broad spectrum of applications. One example is automatic score typesetting, the musical analog of word processing. Almost all score typesetting applications provide a means of automatic generation of a conventional music notation from MIDI data.

In conventional music notation, the onset time of each note is implicitly represented by the cumulative sum of durations of previous notes. Durations are encoded by simple rational numbers (e.g., quarter note, eighth note), consequently all events in music are placed on a discrete grid. So the basic task in MIDI transcription is to associate onset times with discrete grid locations, i.e., quantization.

However, unless the music is performed with mechanical precision, identification of the correct association becomes difficult. This is due to the fact that musicians introduce intentional (and unintentional) deviations from a mechanical prescription. For example timing of events can be deliberately delayed or pushed. Moreover, the tempo can fluctuate by slowing down or accelerating. In fact, such deviations are natural aspects of expressive performance; in the absence of these, music tends to sound rather dull and mechanical. On the other hand, if these deviations are not accounted for during transcription, resulting scores have often very poor quality.

Robust and fast quantization and tempo tracking is also an important requirement for interactive performance systems; applications that “listen” to a performer for generating an accompaniment or improvisation in real time (Raphael, 2001b; Thom, 2000). At last, such models are also useful in musicology for systematic study and characterization of expressive timing by principled analysis of existing performance data.

From a theoretical perspective, simultaneous quantization *and* tempo tracking is a “chicken-and-egg” problem: the quantization depends upon the intended tempo interpretation and the tempo interpretation depends upon the quantization. Apparently, human listeners can resolve this ambiguity (in most cases) without any effort. Even persons without any musical training are able to determine the beat and the tempo very rapidly. However, it is still unclear what precisely constitutes tempo and how it relates to the perception of the beat, rhythmical structure, pitch, style of music etc. Tempo is a perceptual construct and cannot directly be measured in a performance.

The goal of understanding tempo perception has stimulated a significant body of research on the psychological and computational modeling aspects of tempo tracking and beat induction, e.g., see (Desain & Honing, 1994; Large & Jones, 1999; Toiviainen, 1999). These papers assume that events are presented as an onset list. Attempts are also made to deal directly with the audio signal (Goto & Muraoka, 1998; Scheirer, 1998; Dixon & Cambouropoulos, 2000).

Another class of tempo tracking models are developed in the context of interactive performance systems and score following. These models make use of prior knowledge in the form of an annotated score (Dannenberg, 1984; Vercoe & Puckette, 1985). More recently, Raphael (2001b) has demonstrated an interactive real-time system that follows a solo player and schedules accompaniment events according to the player’s tempo interpretation.

Tempo tracking is crucial for quantization, since one can not uniquely quantize onsets without having an estimate of tempo and the beat. The converse, that quantization can help in identification of the correct tempo interpretation has already been noted by Desain and Honing (1991). Here, one defines correct tempo as the one that results in a simpler quantization. However, such a schema has never been fully implemented in practice due to computational complexity of obtaining a perceptually plausible quantization. Hence quantization methods proposed in the literature either estimate the tempo using simple heuristics (Longuet-Higgins, 1987; Pressing & Lawrence, 1993; Agon et al., 1994) or assume that the tempo is known or constant (Desain & Honing, 1991; Cambouropoulos, 2000; Hamanaka et al., 2001).

Our approach to transcription and tempo tracking is from a probabilistic, i.e., Bayesian modeling perspective. In Cemgil et al. (2000), we introduced a probabilistic approach to perceptually realistic quantization. This work also assumed that the tempo was known or was estimated by an

external procedure. For tempo tracking, we introduced a Kalman filter model (Cemgil, Kappen, Desain, & Honing, 2001). In this approach, we modeled the tempo as a smoothly varying hidden state variable of a stochastic dynamical system.

In the current paper, we integrate quantization and tempo tracking. Basically, our model balances score complexity versus smoothness in tempo deviations. The correct tempo interpretation results in a simple quantization and the correct quantization results in a smooth tempo fluctuation. An essentially similar model is proposed recently also by Raphael (2001a). However, Raphael uses an inference technique that only applies for small models; namely when the continuous hidden state is one dimensional. This severely restricts the models one can consider. In the current paper, we survey general and widely used state-of-the-art techniques for inference.

The outline of the paper is as follows: In Section 4.2, we propose a probabilistic model for timing deviations in expressive music performance. Given the model, we will define tempo tracking and quantization as inference of posterior quantities. It will turn out that our model is a switching state space model in which computation of exact probabilities becomes intractable. In Section 4.3, we will introduce approximation techniques based on simulation, namely Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) (Doucet, de Freitas, & Gordon, 2001; Andrieu, de Freitas, Doucet, & Jordan, 2002). Both approaches provide flexible and powerful inference methods that have been successfully applied in diverse fields of applied sciences such as robotics (Fox, Burgard, & Thrun, 1999), aircraft tracking (Gordon, Salmond, & Smith, 1993), computer vision (Isard & Blake, 1996), econometrics (Tanizaki, 2001). Finally we will present simulation results and conclusions.

4.2 Model

Assume that a pianist is improvising and we are recording the exact onset times of each key she presses during the performance. We denote these observed onset times by $y_0, y_1, y_2 \dots y_k \dots y_K$ or more compactly by $y_{0:K}$. We neither have access to a musical notation of the piece nor know the initial tempo she has started her performance with. Moreover, the pianist is allowed to freely change the tempo or introduce expression. Given only onset time information $y_{0:K}$, we wish to find a score $\gamma_{1:K}$ and track her tempo fluctuations $z_{0:K}$. We will refine the meaning of γ and z later.

This problem is apparently ill-posed. If the pianist is allowed to change the tempo arbitrarily it is not possible to assign a “correct” score to a given performance. In other words any performance $y_{0:K}$ can be represented by using a suitable combination of an arbitrary score with an arbitrary tempo trajectory. Fortunately, the Bayes theorem provides an elegant and principled guideline to formulate the problem. Given the onsets $y_{0:K}$, the best score $\gamma_{1:K}$ and tempo trajectory $z_{0:K}$ can be derived from the *posterior* distribution that is given by

$$p(\gamma_{1:K}, z_{0:K} | y_{0:K}) = \frac{1}{p(y_{0:K})} p(y_{0:K} | \gamma_{1:K}, z_{0:K}) p(\gamma_{1:K}, z_{0:K})$$

a quantity, that is proportional to the product of the *likelihood* term $p(y_{0:K} | \gamma_{1:K}, z_{0:K})$ and the *prior* term $p(\gamma_{1:K}, z_{0:K})$.

In rhythm transcription and tempo tracking, the prior encodes our background knowledge about the nature of musical scores and tempo deviations. For example, we can construct a prior that prefers “simple” scores and smooth tempo variations.

The likelihood term relates the tempo and the score to actual observed onset times. In this respect, the likelihood is a model for short time expressive timing deviations and motor errors that are introduced by the performer.

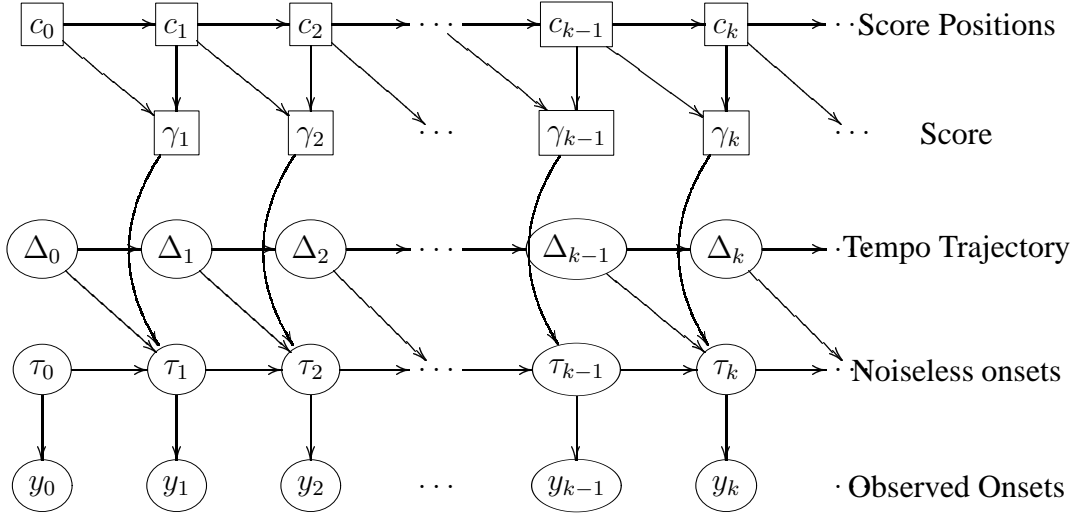


Figure 4.1: Graphical Model. Square and oval nodes correspond to discrete and continuous variables respectively. In the text, we sometimes refer to the continuous hidden variables (τ_k, Δ_k) by z_k . The dependence between γ and c is deterministic. All c, γ, τ and Δ are hidden; only onsets y are observed.

4.2.1 Score prior

To define a score $\gamma_{1:K}$, we first introduce a sequence of *score positions* $c_{0:K}$. A score position c_k specifies the score time of the k 'th onset. We let γ_k denote the interval between score positions of two consecutive onsets

$$\gamma_k = c_k - c_{k-1} \quad (4.1)$$

For example consider the conventional music notation $\downarrow \uparrow$ which encodes the score $\gamma_{1:3} = [1 \ 0.5 \ 0.5]$. Corresponding score positions are $c_{0:3} = [0 \ 1 \ 1.5 \ 2]$.

One simple way of defining a prior distribution on score positions $p(c_k)$ is specifying a table of probabilities for $c_k \bmod 1$ (the fraction of c_k). For example if we wish to allow for scores that have sixteenth notes and triplets, we define a table of probabilities for the states $c \bmod 1 = \{0, 0.25, 0.5, 0.75\} \cup \{0, 0.33, 0.67\}$. Technically, the resulting prior $p(c_k)$ is periodic and improper (since c_k are in principle unbounded so we can not normalize the distribution).

However, if the number of states of $c_k \bmod 1$ is large, it may be difficult to estimate the parameters of the prior reliably. For such situations we propose a “generic” prior as follows: We define the probability, that the k 'th onset gets quantized at location c_k , by $p(c_k) \propto \exp(-\lambda d(c_k))$ where $d(c_k)$ is the number of significant digits in the *binary* expansion of $c_k \bmod 1$. For example $d(1) = 0$, $d(1.5) = 1$, $d(7 + 9/32) = 5$ etc. The positive parameter λ is used to penalize score positions that require more bits to be represented. Assuming that score positions of onsets are independent a-priori, (besides being increasing in k , i.e., $c_k \geq c_{k-1}$), the prior probability of a sequence of score positions is given by $p(c_{0:K}) \propto \exp(-\lambda \sum_{k=0}^K d(c_k))$. We further assume that $c_0 \in [0, 1)$. One can check that such a prior prefers simpler notations, e.g., $p(\uparrow \downarrow \downarrow) < p(\downarrow \uparrow)$. We can generalize this prior to other subdivisions such triplets and quintuplets in Appendix 4.5.

Formally, given a distribution on $c_{0:K}$, the prior of a score $\gamma_{1:K}$ is given by

$$p(\gamma_{1:K}) = \sum_{c_{0:K}} p(\gamma_{1:K} | c_{0:K}) p(c_{0:K}) \quad (4.2)$$

Since the relationship between $c_{0:K}$ and $\gamma_{1:K}$ is deterministic, $p(\gamma_{1:K}|c_{0:K})$ is degenerate for any given $c_{0:K}$, so we have

$$p(\gamma_{1:K}) \propto \exp\left(-\lambda \sum_{k=1}^K d\left(\sum_{k'=1}^k \gamma_{k'}\right)\right) \quad (4.3)$$

One might be tempted to specify a prior directly on $\gamma_{1:K}$ and get rid of $c_{0:K}$ entirely. However, with this simpler approach it is not easy to devise realistic priors. For example, consider a sequence of note durations $[1 \ 1/16 \ 1 \ 1 \ 1 \dots]$. Assuming a factorized prior on γ that penalizes short note durations, this rhythm would have relatively high probability whereas it is quite uncommon in conventional music.

4.2.2 Tempo prior

We represent the tempo in terms of its inverse, i.e., the period, and denote it with Δ . For example a tempo of 120 beats per minute (bpm) corresponds to $\Delta = 60/120 = 0.5$ seconds. At each onset the tempo changes by an unknown amount ζ_{Δ_k} . We assume the change ζ_{Δ_k} is iid with $\mathcal{N}(0, Q_{\Delta})$.² We assume a first order Gauss-Markov process for the tempo

$$\Delta_k = \Delta_{k-1} + \zeta_{\Delta_k} \quad (4.4)$$

Eq. 4.4 defines a distribution over tempo sequences $\Delta_{0:K}$. Given a tempo sequence, the “ideal” or “intended” time τ_k of the next onset is given by

$$\tau_k = \tau_{k-1} + \gamma_k \Delta_{k-1} + \zeta_{\tau_k} \quad (4.5)$$

The noise term ζ_{τ_k} denotes the amount of accentuation (that is deliberately playing a note ahead or back in time) without causing the tempo to be changed. We assume $\zeta_{\tau_k} \sim \mathcal{N}(0, Q_{\tau})$. Ideal onsets and actually observed “noisy” onsets are related by

$$y_k = \tau_k + \epsilon_k \quad (4.6)$$

The noise term ϵ_k models small scale expressive deviations or motor errors in timing of individual notes. In this paper we will assume that ϵ_k has a Gaussian distribution parameterized by $\mathcal{N}(0, R)$.

The initial tempo distribution $p(\Delta_0)$ specifies a range of reasonable tempi and is given by a Gaussian with a broad variance. We assume an uninformative (flat) prior on τ_0 . The conditional independence structure is given by the graphical model in Figure 4.1. Table 4.1 shows a possible realization from the model.

We note that our model is a particular instance of the well known switching state space model (also known as conditionally linear dynamical system, jump Markov linear system, switching Kalman filter) (See, e.g., Bar-Shalom & Li, 1993; Doucet & Andrieu, 2001; Murphy, 2002).

In the following sections, we will sometimes refer use $z_k = (\tau_k, \Delta_k)^T$ and refer to $z_{0:K}$ as a *tempo trajectory*. Given this definition, we can compactly represent Eq. 4.4 and Eq. 4.5 by

$$z_k = \begin{pmatrix} 1 & \gamma_k \\ 0 & 1 \end{pmatrix} z_{k-1} + \zeta_k \quad (4.7)$$

where $\zeta_k = (\zeta_{\tau_k}, \zeta_{\Delta_k})$.

²We denote a (scalar or multivariate) Gaussian distribution $p(\mathbf{x})$ with mean vector μ and covariance matrix P by $\mathcal{N}(\mu, P) \triangleq |2\pi P|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu)^T P^{-1}(\mathbf{x} - \mu))$.




k	0	1	2	3	...
γ_k					...
c_k	0	1/2	3/2	2	...
Δ_k	0.5	0.6	0.7
τ_k	0	0.25	0.85	1.20	...
y_k	0	0.23	0.88	1.24	...

Table 4.1: A possible realization from the model: a ritardando. For clarity we assume $\zeta_\tau = 0$.

4.2.3 Extensions

There are several possible extensions to this basic parameterization. For example, one could represent the period Δ in the logarithmic scale. This warping ensures positivity and seems to be perceptually more plausible since it promotes equal *relative* changes in tempo rather than on an absolute scale (Grubb, 1998; Cemgil et al., 2001). Although the resulting model becomes non-linear, it can be approximated fairly well by an extended Kalman filter (Bar-Shalom & Li, 1993).

A simple random walk model for tempo fluctuations such as in Eq. 4.7 seems not to be very realistic. We would expect the tempo deviations to be more structured and smoother. In our dynamical system framework such smooth deviations can be modeled by increasing the dimensionality of z to include higher order “inertia” variables (Cemgil et al., 2001). For example consider the following model,

$$\begin{pmatrix} \tau_k \\ \Delta_{1,k} \\ \Delta_{2,k} \\ \vdots \\ \Delta_{D-1,k} \end{pmatrix} = \begin{pmatrix} 1 & \gamma_k & \gamma_k & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & & & & \\ \vdots & \vdots & & A & & \\ 0 & 0 & & & & \end{pmatrix} \begin{pmatrix} \tau_{k-1} \\ \Delta_{1,k-1} \\ \Delta_{2,k-1} \\ \vdots \\ \Delta_{D-1,k-1} \end{pmatrix} + \zeta_k \quad (4.8)$$

We choose this particular parameterization because we wish to interpret Δ_1 as the slowly varying “average” tempo and Δ_2 as a temporary change in the tempo. Such a model is useful for situations where the performer fluctuates around an almost constant tempo; a random walk model is not sufficient in this case because it forgets the initial values. Additional state variables $\Delta_3, \dots, \Delta_{D-1}$ act like additional “memory” elements. By choosing the parameter matrix A and noise covariance matrix Q , one can model a rich range of temporal structures in expressive timing deviations.

The score prior can be improved by using a richer model. For example to allow for different time signatures and alternative rhythmic subdivisions, one can introduce additional hidden variables (Cemgil et al., 2000) (See also Appendix 4.5) or use a Markov chain (Raphael, 2001a). Potentially, such extensions make it easier to capture additional structure in musical rhythm (such as “weak” positions are followed more likely by “strong” positions). On the other hand, the number of model parameters rapidly increases and one has to be more cautious in order to avoid overfitting.

For score typesetting, we need to quantize note durations as well, i.e., associate note offsets with score positions. A simple way of accomplishing this is to define an indicator sequence $u_{0:K}$ that identifies whether y_k is an onset ($u_k = 1$) or an offset ($u_k = 0$). Given u_k , we can redefine the observation model as $p(y_k | \tau_k, u_k) = u_k \mathcal{N}(0, R) + (1 - u_k) \mathcal{N}(0, R_{\text{off}})$ where R_{off} is the observation noise associated with offsets. A typical model would have $R_{\text{off}} \gg R$. For $R_{\text{off}} \rightarrow \infty$, the offsets would have no effect on the tempo process. Moreover, since u_k are always observed, this extension requires just a simple lookup.

In principle, one must allow for arbitrary long intervals between onsets, hence γ_k are drawn from an infinite (but discrete) set. In our subsequent derivations, we assume that the number of

possible intervals is fixed a-priori. Given an estimate of z_{k-1} and observation y_k , almost all of the virtually infinite number of choices for γ_k will have almost zero probability and it is easy to identify candidates that would have significant probability mass.

Conceptually, all of the above listed extensions are easy to incorporate into the model and none of them introduces a fundamental computational difficulty to the basic problems of quantization and tempo tracking.

4.2.4 Problem Definition

Given the model, we define rhythm transcription, i.e., quantization as a MAP state estimation problem

$$\begin{aligned}\gamma_{1:K}^* &= \operatorname{argmax}_{\gamma_{1:K}} p(\gamma_{1:K} | y_{0:K}) \\ p(\gamma_{1:K} | y_{0:K}) &= \int dz_{0:K} p(\gamma_{1:K}, z_{0:K} | y_{0:K})\end{aligned}\quad (4.9)$$

and tempo tracking as a filtering problem

$$z_k^* = \operatorname{argmax}_{z_k} \sum_{\gamma_{1:k}} p(\gamma_{1:k}, z_k | y_{0:k}) \quad (4.10)$$

The quantization problem is a smoothing problem: we wish to find the most likely score $\gamma_{1:K}^*$ given all the onsets in the performance. This is useful in “offline” applications such as score typesetting.

For real-time interaction, we need to have an online estimate of the tempo/beat z_k . This information is carried forth by the filtering density $p(\gamma_{1:k}, z_k | y_{0:k})$ in Eq.4.10. Our definition of the best tempo z_k^* as the maximum is somewhat arbitrary. Depending upon the requirements of an application, one can make use of other features of the filtering density. For example, the variance of $\sum_{\gamma_{1:k}} p(\gamma_{1:k}, z_k | y_{0:k})$ can be used to estimate “amount of confidence” in tempo interpretation or $\operatorname{argmax}_{z_k, \gamma_{1:k}} p(\gamma_{1:k}, z_k | y_{0:k})$ to estimate most likely score-tempo pair so far.

Unfortunately, the quantities in Eq. 4.9 and Eq. 4.10 are intractable due to the explosion in the number of mixture components required to represent the exact posterior at each step k (See Figure 4.2). For example, to calculate the exact posterior in Eq. 4.9 we need to evaluate the following expression:

$$p(\gamma_{1:K} | y_{0:K}) = \frac{1}{Z} \int dz_{0:K} p(y_{0:K} | z_{0:K}, \gamma_{1:K}) p(z_{0:K} | \gamma_{1:K}) p(\gamma_{1:K}) \quad (4.11)$$

$$= \frac{1}{Z} p(y_{0:K} | \gamma_{1:K}) p(\gamma_{1:K}) \quad (4.12)$$

where the normalization constant is given by $Z = p(y_{0:K}) = \sum_{\gamma_{1:K}} p(y_{0:K} | \gamma_{1:K}) p(\gamma_{1:K})$. For each trajectory $\gamma_{1:K}$, the integral over $z_{0:K}$ can be computed stepwise in k by the Kalman filter (See appendix 4.5). However, to find the MAP state of Eq. 4.11, we need to evaluate $p(y_{0:K} | \gamma_{1:K})$ independently for each of the exponentially many trajectories. Consequently, the quantization problem in Eq. 4.9 can only be solved approximately.

For accurate approximation, we wish to exploit any inherent independence structure of the exact posterior. Unfortunately, since z and c are integrated over, all γ_k become coupled and in general $p(\gamma_{1:K} | y_{0:K})$ does not possess any conditional independence structure (e.g., a Markov chain) that would facilitate efficient calculation. Consequently, we will resort to numerical approximation techniques.

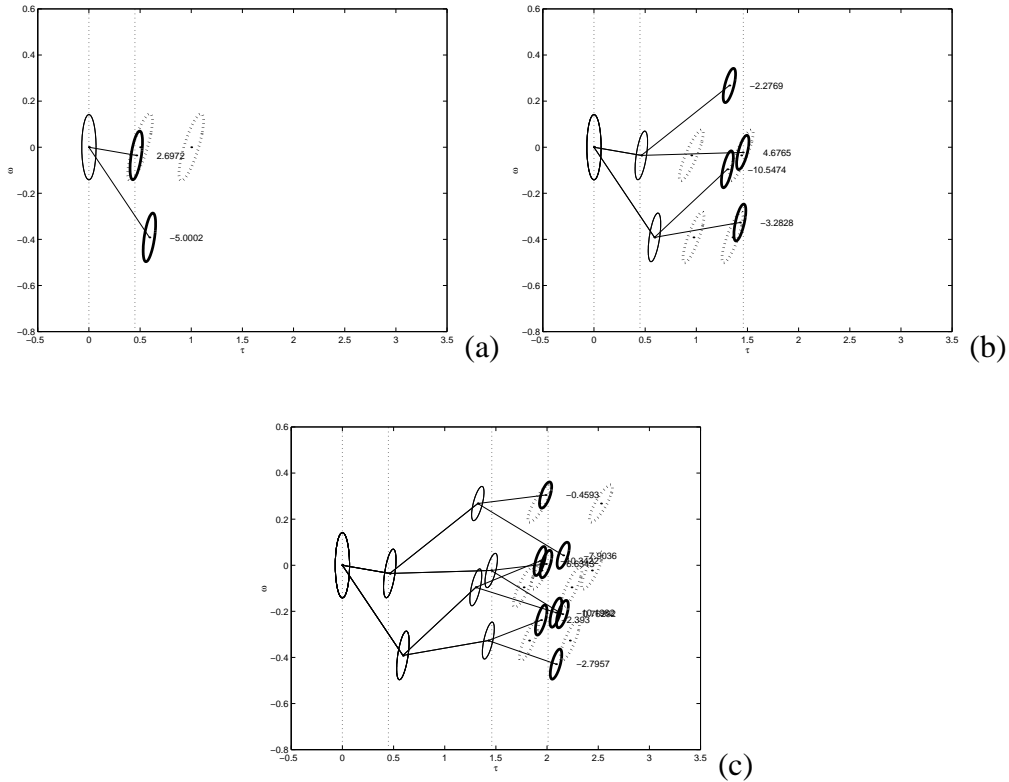


Figure 4.2: Example demonstrating the explosion of the number of components to represent the exact posterior. Ellipses denote the conditional marginals $p(\tau_k, \omega_k | c_{0:k}, y_{0:k})$. (We show the period in logarithmic scale where $\omega_k = \log_2 \Delta_k$). In this toy example, we assume that a score consists only of notes of length \uparrow and \downarrow , i.e., γ_k can be either $1/2$ or 1 . **(a)** We start with a unimodal posterior $p(\tau_0, \omega_0 | c_0, y_0)$, e.g., a Gaussian centered at $(\tau, \omega) = (0, 0)$. Since we assume that a score can only consist of eight- and quarter notes, i.e., $\gamma_k \in \{1/2, 1\}$, the predictive distribution $p(\tau_1, \omega_1 | c_{0:1}, y_0)$ is bimodal where the modes are centered at $(0.5, 0)$ and $(1, 0)$ respectively (shown with a dashed contour line). Once the next observation y_1 is observed (shown with a dashed vertical line around $\tau = 0.5$), the predictive distribution is updated to yield $p(\tau_1, \omega_1 | c_{0:1}, y_{0:1})$. The numbers denote the respective log-posterior weight of each mixture component. **(b)** The predictive distribution $p(\tau_2, \omega_2 | c_{0:1}, y_{0:1})$ at step $k = 2$ has now 4 modes, two for each component of $p(\tau_1, \omega_1 | c_{0:1}, y_{0:1})$. **(c)** The number of components grows exponentially with k .

4.3 Monte Carlo Simulation

Consider a high dimensional probability distribution

$$p(\mathbf{x}) = \frac{1}{Z} p^*(\mathbf{x}) \quad (4.13)$$

where the normalization constant $Z = \int d\mathbf{x} p^*(\mathbf{x})$ is not known but $p^*(\mathbf{x})$ can be evaluated at any particular \mathbf{x} . Suppose we want to estimate the expectation of a function $f(\mathbf{x})$ under the distribution $p(\mathbf{x})$ denoted as

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} = \int d\mathbf{x} f(\mathbf{x}) p(\mathbf{x})$$

e.g., the mean of \mathbf{x} under $p(\mathbf{x})$ is given by $\langle \mathbf{x} \rangle$. The intractable integration can be approximated by an average if we can find N points $\mathbf{x}^{(i)}$, $i = 1 \dots N$ from $p(\mathbf{x})$

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^{(i)}) \quad (4.14)$$

When $\mathbf{x}^{(i)}$ are generated by independently sampling from $p(\mathbf{x})$, it can be shown that as N approaches infinity, the approximation becomes exact.

However, generating independent samples from $p(\mathbf{x})$ is a difficult task in high dimensions but it is usually easier to generate *dependent* samples, that is we generate $\mathbf{x}^{(i+1)}$ by making use of $\mathbf{x}^{(i)}$. It is somewhat surprising, that even if $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ are correlated (and provided ergodicity conditions are satisfied), Eq. 4.14 remains still valid and estimated quantities converge to their true values when number of samples N goes to infinity.

A sequence of dependent samples $\mathbf{x}^{(i)}$ is generated by using a Markov chain that has the stationary distribution $p(\mathbf{x})$. The chain is defined by a collection of transition probabilities, i.e., a *transition kernel* $T(\mathbf{x}^{(i+1)}|\mathbf{x}^{(i)})$. The definition of the kernel is implicit, in the sense that one defines a procedure to generate the $\mathbf{x}^{(i+1)}$ given $\mathbf{x}^{(i)}$. The *Metropolis* algorithm (Metropolis & Ulam, 1949; Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) provides a simple way of defining an ergodic kernel that has the desired stationary distribution $p(\mathbf{x})$. Suppose we have a sample $\mathbf{x}^{(i)}$. A candidate \mathbf{x}' is generated by sampling from a symmetric proposal distribution $q(\mathbf{x}'|\mathbf{x}^{(i)})$ (for example a Gaussian centered at $\mathbf{x}^{(i)}$). The candidate \mathbf{x}' is accepted as the next sample $\mathbf{x}^{(i+1)}$ if $p(\mathbf{x}') > p(\mathbf{x}^{(i)})$. If \mathbf{x}' has a lower probability, it can be still accepted, but only with probability $p(\mathbf{x}')/p(\mathbf{x}^{(i)})$. The algorithm is initialized by generating the first sample $\mathbf{x}^{(0)}$ according to an (arbitrary) proposal distribution.

However for a given transition kernel T , it is hard to assess the time required to converge to the stationary distribution so in practice one has to run the simulation until a very large number of samples have been obtained, (see e.g., Roberts & Rosenthal, 1998). The choice of the proposal distribution q is also very critical. A poor choice may lead to the rejection of many candidates \mathbf{x}' hence resulting in a very slow convergence to the stationary distribution.

For a large class of probability models, where the full posterior $p(\mathbf{x})$ is intractable, one can still efficiently compute marginals of form $p(x_k|\mathbf{x}_{-k})$, $\mathbf{x}_{-k} = x_1 \dots x_{k-1}, x_{k+1}, \dots x_K$ exactly. In this case one can apply a more specialized Markov chain Monte Carlo (MCMC) algorithm, the *Gibbs sampler* given below.

1. Initialize $x_{1:K}^{(0)}$ by sampling from a proposal $q(x_{1:K})$
2. For $i = 0 \dots N - 1$

- For $k = 1, \dots, K$, Sample

$$x_k^{(i+1)} \sim p(x_k | x_{1:k-1}^{(i+1)}, x_{k+1:K}^{(i)}) \quad (4.15)$$

In contrast to the Metropolis algorithm, where the new candidate is a vector \mathbf{x}' , the Gibbs sampler uses the exact marginal $p(x_k | \mathbf{x}_{-k})$ as the proposal distribution. At each step, the sampler updates only one coordinate of the current state \mathbf{x} , namely x_k , and the new candidate is guaranteed to be accepted.

Note that, in principle we don't need to sample x_k sequentially, i.e., we can choose k randomly provided that each slice is visited equally often in the limit. However, a deterministic scan algorithm where $k = 1, \dots, K$, provides important time savings in the type of models that we consider here.

4.3.1 Simulated Annealing and Iterative Improvement

Now we shift our focus from sampling to MAP state estimation. In principle, one can use the samples generated by any sampling algorithm (Metropolis-Hastings or Gibbs) to estimate the MAP state \mathbf{x}^* of $p(\mathbf{x})$ by $\underset{i=1:N}{\operatorname{argmax}} p(\mathbf{x}^{(i)})$. However, unless the posterior is very much concentrated around the MAP state, the sampler may not visit \mathbf{x}^* even though the samples $\mathbf{x}^{(i)}$ are obtained from the stationary distribution. In this case, the problem can be simply reformulated to sample not from $p(\mathbf{x})$ but from a distribution that is concentrated at local maxima of $p(\mathbf{x})$. One such class of distributions are given by $p_{\rho_j}(\mathbf{x}) \propto p(\mathbf{x})^{\rho_j}$. A sequence of exponents $\rho_1 < \rho_2 < \dots < \rho_j < \dots$ is called to be a *cooling schedule* or *annealing schedule* owing to the inverse temperature interpretation of ρ_j in statistical mechanics, hence the name *Simulated Annealing* (SA) (Aarts & van Laarhoven, 1985). When $\rho_j \rightarrow \infty$ sufficiently slowly in j , the cascade of MCMC samplers each with the stationary distribution $p_{\rho_j}(\mathbf{x})$ is guaranteed (in the limit) to converge to the global maximum of $p(\mathbf{x})$. Unfortunately, for this convergence result to hold, the cooling schedule must go very slowly (in fact, logarithmically) to infinity. In practice, faster cooling schedules must be employed.

Iterative improvement (II) (Aarts & van Laarhoven, 1985) is a heuristic simulated annealing algorithm with a very fast cooling schedule. In fact, $\rho_j = \infty$ for all j . The eventual advantage of this greedy algorithm is that it converges in a few iterations to a local maximum. By restarting many times from different initial configurations \mathbf{x} , one hopes to find different local maxima of $p(\mathbf{x})$ and eventually visit the MAP state \mathbf{x}^* . In practice, by using the II heuristic one may find better solutions than SA for a limited computation time.

From an implementation point of view, it is trivial to convert MCMC code to SA (or II) code. For example, consider the Gibbs sampler. To implement SA, we need to construct a cascade of Gibbs samplers, each with stationary distribution $p(\mathbf{x})^{\rho_j}$. The exact one time slice marginal of this distribution is $p(x_k | \mathbf{x}_{-k})^{\rho_j}$. So, SA just samples from the actual (temperature=1) marginal $p(x_k | \mathbf{x}_{-k})$ raised to a power ρ_j .

4.3.2 The Switching State Space Model and MAP Estimation

To solve the rhythm quantization problem, we need to calculate the MAP state of the posterior in Eq. 4.11

$$p(\gamma_{1:K} | y_{0:K}) \propto p(\gamma_{1:K}) \int dz_{0:K} p(y_{0:K} | z_{0:K}, \gamma_{1:K}) p(z_{0:K} | \gamma_{1:K}) \quad (4.16)$$

This is a combinatorial optimization problem: we seek the maximum of a function $p(\gamma_{1:K} | y_{0:K})$ that associates a number with each of the discrete configurations $\gamma_{1:K}$. Since it is not feasible to

visit all of the exponentially many configurations to find the maximizing configuration $\gamma_{1:K}^*$, we will resort to stochastic search algorithms such as simulated annealing (SA) and iterative improvement (II). Due to the strong relationship between the Gibbs sampler and SA (or II), we will first review the Gibbs sampler for the switching state space model.

The first important observation is that, conditioned on $\gamma_{1:K}$, the model becomes a linear state space model and the integration on $z_{0:K}$ can be computed analytically using Kalman filtering equations. Consequently, one can sample only $\gamma_{1:K}$ and integrate out z . The analytical marginalization, called *Rao-Blackwellization* (Casella & Robert, 1996), improves the efficiency of the sampler (e.g., see Doucet, de Freitas, Murphy, & Russell, 2000a).

Suppose now that each switch variable γ_k can have S distinct states and we wish to generate N samples (i.e trajectories) $\{\gamma_{1:K}^{(i)}, i = 1 \dots N\}$. A naive implementation of the Gibbs sampler requires that at each step k we run the Kalman filter S times on the whole observation sequence $y_{0:K}$ to compute the proposal $p(\gamma_k | \gamma_{1:k-1}^{(i)}, \gamma_{k+1:K}^{(i-1)}, y_{0:K})$. This would result in an algorithm of time complexity $O(NK^2S)$ that is prohibitively slow when K is large. Carter and Kohn (1996) have proposed a much more time efficient deterministic scan Gibbs sampler that circumvents the need to run the Kalman filtering equations at each step k on the whole observation sequence $y_{0:K}$. See also (Doucet & Andrieu, 2001; Murphy, 2002).

The method is based on the observation that the proposal distribution $p(\gamma_k | \cdot)$ can be factorized as a product of terms that either depend on past observations $y_{0:k}$ or the future observations $y_{k+1:K}$. So the contribution of the future can be computed a-priori by a backward filtering pass. Subsequently, the proposal is computed and samples $\gamma_k^{(i)}$ are generated during the forward pass. The sampling distribution is given by

$$p(\gamma_k | \gamma_{-k}, y_{0:K}) \propto p(\gamma_k | \gamma_{-k}) p(y_{0:K} | \gamma_{1:K}) \quad (4.17)$$

where the first term is proportional to the joint prior $p(\gamma_k | \gamma_{-k}) \propto p(\gamma_k, \gamma_{-k})$. The second term can be decomposed as

$$p(y_{0:K} | \gamma_{1:K}) = \int dz_k p(y_{k+1:K} | y_{0:k}, z_k, \gamma_{1:K}) p(y_{0:k}, z_k | \gamma_{1:K}) \quad (4.18)$$

$$= \int dz_k p(y_{k+1:K} | z_k, \gamma_{k+1:K}) p(y_{0:k}, z_k | \gamma_{1:k}) \quad (4.19)$$

Both terms are (unnormalized) Gaussian potentials hence the integral can be evaluated analytically. The term $p(y_{k+1:K} | z_k, \gamma_{k+1:K})$ is an unnormalized Gaussian potential in z_k and can be computed by backwards filtering. The second term is just the filtering distribution $p(z_k | y_{0:k}, \gamma_{1:k})$ scaled by the likelihood $p(y_{0:k} | \gamma_{1:k})$ and can be computed during forward filtering. The outline of the algorithm is given below, see the appendix 4.5 for details.

1. Initialize $\gamma_{1:K}^{(0)}$ by sampling from a proposal $q(\gamma_{1:K})$
2. For $i = 1 \dots N$
 - For $k = K - 1, \dots, 0$,
 - Compute $p(y_{k+1:K} | z_k, \gamma_{k+1:K}^{(i-1)})$
 - For $k = 1, \dots, K$,
 - For $s = 1 \dots S$
 - * Compute the proposal

$$p(\gamma_k = s | \cdot) \propto p(\gamma_k = s, \gamma_{-k}) \int dz_k p(y_{0:k}, z_k | \gamma_{1:k-1}^{(i)}, \gamma_k = s) p(y_{k+1:K} | z_k, \gamma_{k+1:K}^{(i-1)})$$

- Sample $\gamma_k^{(i)}$ from $p(\gamma_k|\cdot)$

The resulting algorithm has a time complexity of $O(NKS)$, an important saving in terms of time. However, the space complexity increases from $O(1)$ to $O(K)$ since expectations computed during the backward pass need to be stored.

At each step, the Gibbs sampler generates a sample from a single time slice k . In certain types of “sticky” models, such as when the dependence between γ_k and γ_{k+1} is strong, the sampler may get stuck in one configuration, moving very rarely. This is due to the fact that most singleton flips end up in low probability configurations due to the strong dependence between adjacent time slices. As an example, consider the quantization model and two configurations $[\dots \gamma_k, \gamma_{k+1} \dots] = [\dots 1, 1 \dots]$ and $[\dots 3/2, 1/2 \dots]$. By updating only a single slice, it may be difficult to move between these two configurations. Consider an intermediate configuration $[\dots 3/2, 1 \dots]$. Since the duration $(\gamma_k + \gamma_{k+1})$ increases, all future score positions $c_{k:K}$ are shifted by $1/2$. That may correspond to a score that is heavily penalized by the prior, thus “blocking” the path.

To allow the sampler move more freely, i.e., to allow for more global jumps, one can sample from L slices jointly. In this case the proposal distribution takes the form

$$p(\gamma_{k:k+L-1}|\cdot) \propto p(\gamma_{k:k+L-1}, \boldsymbol{\gamma}_{-(k:k+L-1)}) \times \int dz_{k+L-1} p(y_{0:k+L-1}, z_{k+L-1} | \gamma_{1:k-1}^{(i)}, \gamma_{k:k+L-1}) p(y_{k+L:K} | z_{k+L-1}, \gamma_{k+L:K}^{(i-1)})$$

Similar to the one slice case, terms under the integral are unnormalized Gaussian potentials (on z_{k+L-1}) representing the contribution of past and future observations. Since $\gamma_{k:k+L-1}$ has S^L states, the resulting time complexity for generating N samples is $O(NKS^L)$, thus in practice L must be kept rather small. One remedy would be to use a Metropolis-Hastings algorithm with a heuristic proposal distribution $q(\gamma_{k:k+L-1}|y_{0:K})$ to circumvent exact calculation, but it is not obvious how to construct such a q .

One other shortcoming of the Gibbs sampler (and related MCMC methods) is that the algorithm in its standard form is inherently offline; we need to have access to all of the observations $y_{0:K}$ to start the simulation. For certain applications, e.g., automatic score typesetting, a batch algorithm might be still feasible. However in scenarios that require real-time interaction, such as in interactive music performance or tempo tracking, online methods must be used.

4.3.3 Sequential Monte Carlo

Sequential Monte Carlo, a.k.a. particle filtering, is a powerful alternative to MCMC for generating samples from a target posterior distribution. SMC is especially suitable for application in dynamical systems, where observations arrive sequentially.

The basic idea in SMC is to represent the posterior $p(x_{0:k-1}|y_{0:k-1})$ at time $k-1$ by a (possibly weighted) set of samples $\{x_{0:k-1}^{(i)}, i = 1 \dots N\}$ and extend this representation to $\{(x_{0:k-1}^{(i)}, x_k^{(i)}), i = 1 \dots N\}$ when the observation y_k becomes available at time k . The common practice is to use importance sampling.

Importance Sampling

Consider again a high dimensional probability distribution $p(\mathbf{x}) = p^*(\mathbf{x})/Z$ with an unknown normalization constant. Suppose we are given a *proposal* distribution $q(\mathbf{x})$ that is close to $p(\mathbf{x})$ such that high probability regions of both distributions fairly overlap. We generate independent

samples, i.e., *particles*, $\mathbf{x}^{(i)}$ from the proposal such that $q(\mathbf{x}) \approx \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}^{(i)})/N$. Then we can approximate

$$p(\mathbf{x}) = \frac{1}{Z} \frac{p^*(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \quad (4.20)$$

$$\approx \frac{1}{Z} \frac{p^*(\mathbf{x})}{q(\mathbf{x})} \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}^{(i)}) \quad (4.21)$$

$$\approx \sum_{i=1}^N \frac{w^{(i)}}{\sum_{j=1}^N w^{(j)}} \delta(\mathbf{x} - \mathbf{x}^{(i)}) \quad (4.22)$$

where $w^{(i)} = p^*(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$ are the *importance weights*. One can interpret $w^{(i)}$ as correction factors to compensate for the fact that we have sampled from the “incorrect” distribution $q(\mathbf{x})$. Given the approximation in Eq.4.22 we can estimate expectations by weighted averages

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} \approx \sum_{i=1}^N \tilde{w}^{(i)} f(\mathbf{x}^{(i)}) \quad (4.23)$$

where $\tilde{w}^{(i)} = w^{(i)} / \sum_{j=1}^N w^{(j)}$ are the *normalized importance weights*.

Sequential Importance Sampling

Now we wish to apply importance sampling to the dynamical model

$$p(x_{0:K} | y_{0:K}) \propto \prod_{k=0}^K p(y_k | x_k) p(x_k | x_{0:k-1}) \quad (4.24)$$

where $x = \{z, \gamma\}$. In principle one can naively apply standard importance sampling by using an arbitrary proposal distribution $q(x_{0:K})$. However finding a good proposal distribution can be hard if $K \gg 1$. The key idea in *sequential importance sampling* is the sequential construction of the proposal distribution, possibly using the available observations $y_{0:k}$, i.e.,

$$q(x_{0:K} | y_{0:K}) = \prod_{k=0}^K q(x_k | x_{0:k-1}, y_{0:k})$$

Given a sequentially constructed proposal distribution, one can compute the importance weight recursively as

$$w_k^{(i)} = \frac{p^*(x_{0:k}^{(i)} | y_{0:k})}{q(x_{0:k}^{(i)} | y_{0:k})} = \frac{p(y_k | x_k^{(i)}) p(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{0:k-1}) p(y_{0:k-1} | x_{0:k-1}^{(i)}) p(x_{0:k-1}^{(i)})}{q(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{0:k}) q(x_{0:k-1}^{(i)} | y_{0:k-1})} \quad (4.25)$$

$$= \frac{p(y_k | x_k^{(i)}) p(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{0:k-1})}{q(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{0:k})} w_{k-1}^{(i)} \quad (4.26)$$

The sequential update schema is potentially more accurate than naive importance sampling since at each step k , one can generate a particle from a fairly accurate proposal distribution that takes the current observation y_k into account. A natural choice for the proposal distribution is the filtering distribution given as

$$q(x_k | x_{0:k-1}^{(i)}, y_{0:k}) = p(x_k | x_{0:k-1}^{(i)}, y_{0:k}) \quad (4.27)$$

In this case the weight update rule in Eq. 4.26 simplifies to

$$w_k^{(i)} = p(y_k | x_{0:k-1}^{(i)}) w_{k-1}^{(i)}$$

In fact, provided that the proposal distribution q is constructed sequentially and past sampled trajectories are not updated, the filtering distribution is the optimal choice in the sense of minimizing the variance of importance weights $w^{(i)}$ (Doucet, Godsill, & Andrieu, 2000b). Note that Eq. 4.27 is identical to the proposal distribution used in Gibbs sampling at $k = K$ (Eq 4.15). At $k < K$, the SMC proposal does not take future observations into account; so we introduce discount factors w_k to compensate for sampling from the wrong distribution.

Selection

Unfortunately, the sequential importance sampling may be degenerate, in fact, it can be shown that the variance of $w_k^{(i)}$ increases with k . In practice, after a few iterations of the algorithm, only one particle has almost all of the probability mass and most of the computation time is wasted for updating particles with negligible probability.

To avoid the undesired degeneracy problem, several heuristic approaches are proposed in the literature. The basic idea is to duplicate or discard particles according to their normalized importance weights. The selection procedure can be deterministic or stochastic. Deterministic selection is usually greedy; one chooses N particles with the highest importance weights. In the stochastic case, called *resampling*, particles are drawn with a probability proportional to their importance weight $w_k^{(i)}$. Recall that normalized weights $\{\tilde{w}_k^{(i)}, i = 1 \dots N\}$ can be interpreted as a discrete distribution on particle labels (i).

4.3.4 SMC for the Switching State Space Model

The SIS algorithm can be directly applied to the switching state space model by sampling directly from $x_k = (z_k, \gamma_k)$. However, the particulate approximation can be quite poor if z is high dimensional. Hence, too many particles may be needed to accurately represent the posterior.

Similar to the MCMC methods introduced in the previous section, efficiency can be improved by analytically integrating out $z_{0:k}$ and only sampling from $\gamma_{1:k}$. This form of Rao-Blackwellization is reported to give superior results when compared to standard particle filtering where both γ and z are sampled jointly (Chen & Liu, 2000; Doucet et al., 2000b). The improvement is perhaps not surprising, since importance sampling performs best when the sampled space is low dimensional.

The algorithm has an intuitive interpretation in terms of a randomized breadth first tree search procedure: at each new step k , we expand N kernels to obtain $S \times N$ new kernels. Consequently, to avoid explosion in the number of branches, we select N out of $S \times N$ branches proportional to the likelihood, See Figure 4.3. The derivation and technical details of the algorithm are given in the Appendix 4.5.

The tree search interpretation immediately suggests a deterministic version of the algorithm where one selects (without replacement) the N branches with highest weight. We will refer to this method as a *greedy filter* (GF). The method is also known as *split-track* filter (Chen & Liu, 2000) and is closely related to Multiple Hypothesis Tracking (MHT) (Bar-Shalom & Fortmann, 1988). One problem with the greedy selection schema of GF is the loss of particle diversity. Even if the particles are initialized to different locations in z_0 , (e.g., to different initial tempi), mainly due to the discrete nature of the state space of γ_k , most of the particles become identical after a few steps k . Consequently, results can not be improved by increasing the number of particles N . Nevertheless, when only very few particles can be used, say e.g., in a real time application, GF may still be a viable choice.

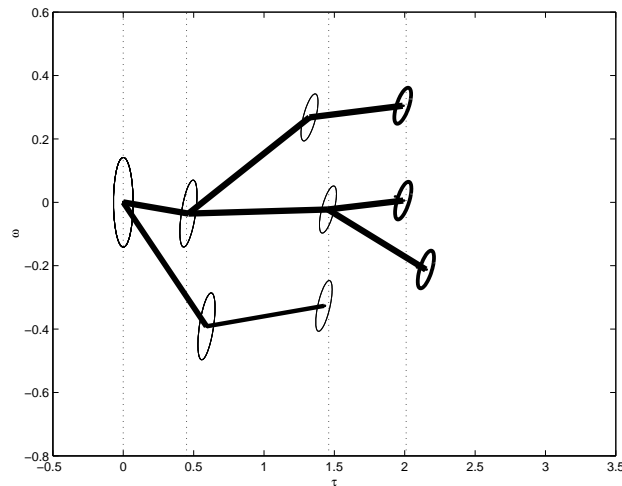


Figure 4.3: Outline of the algorithm. The ellipses correspond to the conditionals $p(z_k | \gamma_k^{(i)}, y_{0:k})$. Vertical dotted lines denote the observations y_k . At each step k , particles with low likelihood are discarded. Surviving particles are linked to their parents.

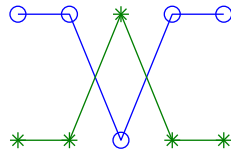


Figure 4.4: A hypothetical situation where neither of the two particles $\gamma_{1:5}^{(i)}$ is optimal. We would obtain eventually a higher likelihood configuration by interchanging γ_3 between particles.

4.3.5 SMC and estimation of the MAP trajectory

Like MCMC, SMC is a sampling method. Hence comments made in Section 4.3.1 about the eventual suboptimality of estimating the MAP trajectory from particles as $\arg \max p(\gamma_{1:K}^{(i)} | y_{0:K})$ also apply here. An hypothetical situation is shown in figure 4.4.

One obvious solution is to employ the SA “trick” and raise the proposal distribution to a power $p(\gamma_k | \cdot)^\gamma$. However, such a proposal will be peaked on a very few γ at each time slice. Consequently, most of the particles will become identical in time and the algorithm eventually degenerates to greedy filtering.

An algorithm for estimating the MAP trajectory from a set of SMC samples is recently proposed in the literature (Godsill, Doucet, & West, 2001). The algorithm relies on the observation that once the particles $x_k^{(i)}$ are sampled during the forward pass, one is left with a discrete distribution defined on the (discrete) support $X_{1:K} = \bigotimes_{k=1}^K X_k$. Here X_k denotes the support of the filtering distribution at time k and \bigotimes is the Cartesian product between sets. Formally, X_k is the set of *distinct* samples at time k and is given by $X_k = \bigcup_i \{x_k^{(i)}\}$.

The distribution $p(X_{1:K}|y_{1:K})^3$ is Markovian because the original state transition model is Markovian, i.e., the posterior can be represented exactly by

$$p(X_{1:K}|y_{1:K}) \propto \prod_{k=1}^K p(y_k|X_k)p(X_k|X_{k-1})$$

Consequently, one can find the best MAP trajectory $\arg \max p(X_{1:K})$ by using an algorithm that is analogous to the Viterbi algorithm for hidden Markov models (Rabiner, 1989).

However, this idea does not carry directly to the case when one applies Rao-Blackwellization. In general, when a subset of the hidden variables is integrated out, all time slices of the posterior $p(\Gamma_{1:K}|y_{1:k})$ are coupled, where $\Gamma_{1:K} = \bigotimes_{k=1}^K \Gamma_k$ and $\Gamma_k = \bigcup_i \{\gamma_k^{(i)}\}$. One can still employ a chain approximation and run Viterbi, (e.g., Cemgil & Kappen, 2002), but this does not guarantee to find $\arg \max p(\Gamma_{1:K}|y_{1:k})$.

On the other hand, because $\gamma_k^{(i)}$ are drawn from a discrete set, several particles become identical so Γ_k has usually a small cardinality when compared to the number of particles N . Consequently, it becomes feasible to employ SA or II on the reduced state space $\Gamma_{1:K}$; possibly using a proposal distribution that extends over several time slices L .

In practice, for finding the MAP solution from the particle set $\{\gamma_{1:K}^{(i)}, i = 1 \dots N\}$, we propose to find the best trajectory $i^* = \arg \max_i p(y_{0:K}|\gamma_{1:K}^{(i)})p(\gamma_{1:K}^{(i)})$ and apply iterative improvement starting from the initial configuration $\gamma_{1:K}^{(i^*)}$.

4.4 Simulations

We have compared the inference methods in terms of the quality of the solution and execution time. The tests are carried out both on artificial and real data.

Given the true notation $\gamma_{1:K}^{\text{true}}$, we measure the quality of a solution in terms of the log-likelihood difference

$$\Delta \mathcal{L} = \log \frac{p(y_{0:K}|\gamma_{1:K})p(\gamma_{1:K})}{p(y_{0:K}|\gamma_{1:K}^{\text{true}})p(\gamma_{1:K}^{\text{true}})}$$

and in terms of *edit distance*

$$e(\gamma_{1:K}) = \sum_{k=1}^K (1 - \delta(\gamma_k - \gamma_k^{\text{true}}))$$

The edit distance $e(\gamma_{1:K})$ gives simply the number of notes that are quantized wrongly.

4.4.1 Artificial data: Clave pattern

The synthetic example is a repeating ‘‘son-clave’’ pattern $\mathbb{1} \dot{\downarrow} \downarrow \downarrow \dot{\downarrow} \downarrow \downarrow \dot{\downarrow} \downarrow \downarrow \mathbb{1} (c = [1, 2, 4, 5.5, 7 \dots])$ with fluctuating tempo. We repeat the pattern 6 times and obtain a score $\gamma_{1:K}$ with $K = 30$.

Such syncopated rhythms are usually hard to transcribe and make it difficult to track the tempo even for experienced human listeners. Moreover, since onsets are absent at prominent beat locations, standard beat tracking algorithms usually loose track.

³By a slight abuse of notation we use the symbol X_k both as a set and as a general element when used in the argument of a density, $p(y_k|X_k)$ means $p(y_k|x_k)$ s.t. $x_k \in X_k$

Given score $\gamma_{1:K}$, we have generated 100 observation sequences $y_{0:K}$ by sampling from the tempo model in Eq. 4.7. We have parameterized the observation noise variance⁴ as $Q = \gamma_k Q_a + Q_b$. In this formulation, the variance depends on the length of the interval between consecutive onsets; longer notes in the score allow for more tempo and timing fluctuation. For the tests on the clave example we have not used a prior model that reflects true source statistics, instead, we have used the generic prior model defined in Section 4.2.1 with $\lambda = 1$.

All the example cases are sampled from the same score (clave pattern). However, due to the use of the generic prior (that does not capture the exact source statistics well) and a relatively broad noise model, the MAP trajectory $\gamma_{1:K}^*$ given $y_{0:K}$ is not always identical to the original clave pattern. For the i 'th example, we have defined the ‘‘ground truth’’ $\gamma_{1:K}^{\text{true},i}$ as the highest likelihood solution found using any sampling technique during any independent run. Although this definition of the ground truth introduces some bias, we have found this exercise more realistic as well as more discriminative among various methods when compared to, e.g., using a dataset with essentially shorter sequences where the exact MAP trajectory can be computed by exhaustive enumeration. The wish to stress that the main aim of the simulations on synthetic dataset is to compare effectiveness of different inference techniques; we postpone the actual test whether the model is a good one to our simulations on real data.

We have tested the MCMC methods, namely Gibbs sampling (Gibbs), simulated annealing (SA) and iterative improvement (II) with one and two time slice optimal proposal and for 10 and 50 sweeps. For each onset y_k , the optimal proposal $p(\gamma_k|\cdot)$ is computed always on a fixed set, $\Gamma = \{0, 1/4, 2/4 \dots 3\}$. Figure 4.6 shows a typical run of MCMC.

Similarly, we have implemented the SMC for $N = \{1, 5, 10, 50, 100\}$ particles. The selection schema was random drawing from the optimal proposal $p(\gamma_k|\cdot)$ computed using one or two time slices. Only in the special case of greedy filtering (GF), i.e., when $N = 1$, we have selected the switch with maximum probability. An example run is shown in Figure 4.5.

We observe that on average SMC results are superior to MCMC (Figure 4.7). We observe that, increasing the number of sweeps for MCMC does not improve the solution significantly. On the other hand, increasing the number of particles seems to improve the quality of the SMC solution monotonically. Moreover, the results suggest that sampling from two time slices jointly (with the exception of SA) does not have a big effect. GF outperforms a particle filter with 5 particles that draws randomly from the proposal. That suggests that for PF with a small number of particles N , it may be desirable to use a hybrid selection schema that selects the particle with maximum weight automatically and randomly selects the remaining $N - 1$.

We compare inference methods in terms of execution time and the quality of solutions (as measured by edit distance). As Figure 4.8 suggests, using a two slice proposal is not justified. Moreover it seems that for comparable computational effort, SMC tends to outperform all MCMC methods.

4.4.2 Real Data: Beatles

We evaluate the performance of the model on polyphonic piano performances. 12 pianists were invited to play two Beatles songs, Michelle and Yesterday. Both pieces have a relatively simple rhythmic structure with ample opportunity to add expressiveness by fluctuating the tempo. The original score is shown in Figure 4.9(a). The subjects had different musical education and background: four professional jazz players, four professional classical performers and four amateur classical pianists. Each arrangement had to be played in three tempo conditions, three repetitions per tempo condition. The tempo conditions were normal, slow and fast tempo (all in a musically

⁴The noise covariance parameters were $R = 0.02^2$, $Q_a = 0.06^2 I$ and $Q_b = 0.02^2 I$. I is a 2×2 identity matrix.

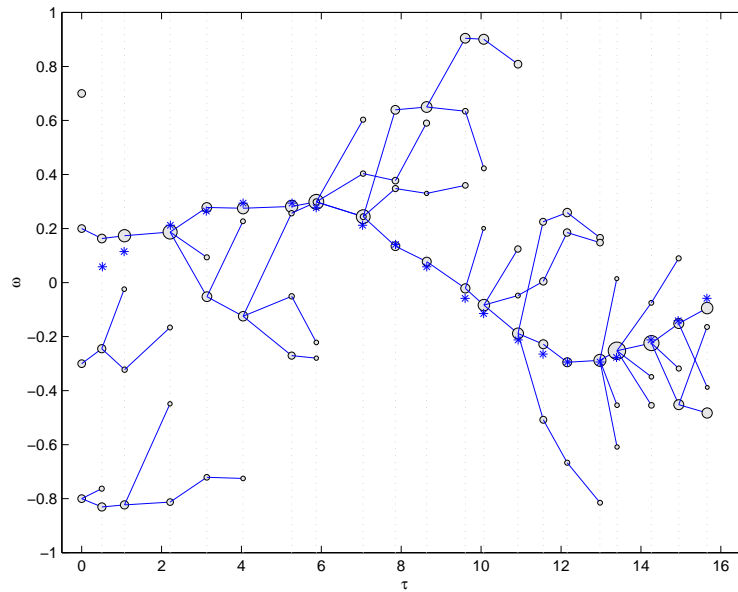


Figure 4.5: Particle filtering on clavichord example with 4 particles. Each circle denotes the mean $(\tau_k^{(n)}, \omega_k^{(n)})$ where $\omega_k^{(n)} = \log_2 \Delta_k$. The diameter of each particle is proportional to the normalized importance weight at each generation. '*' denote the true (τ, ω) pairs; here we have modulated the tempo deterministically according to $\omega_k = 0.3 \sin(2\pi c_k/32)$, observation noise variance is $R = 0.025^2$.

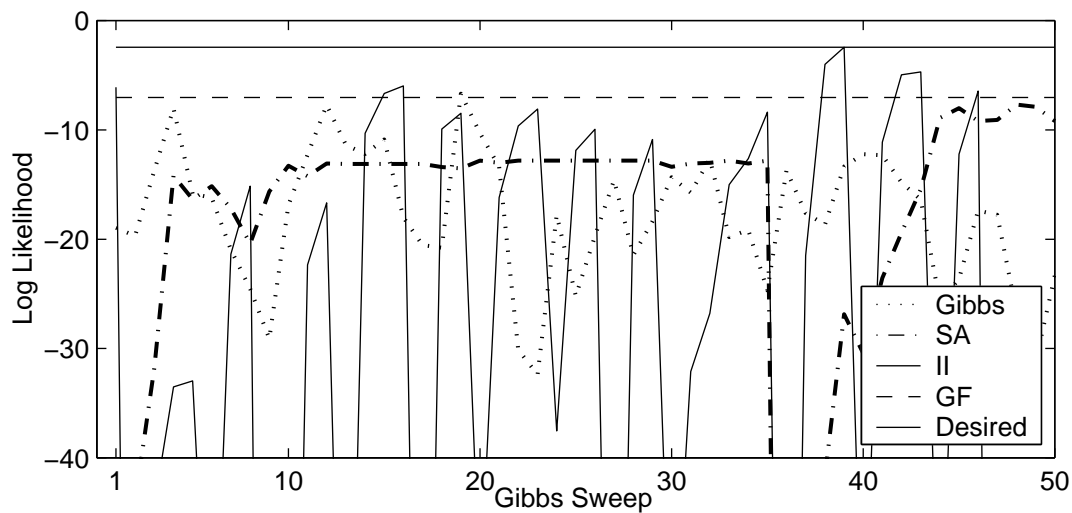
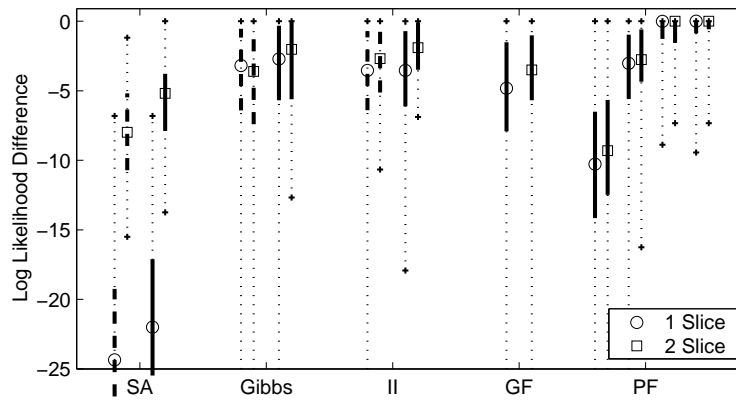
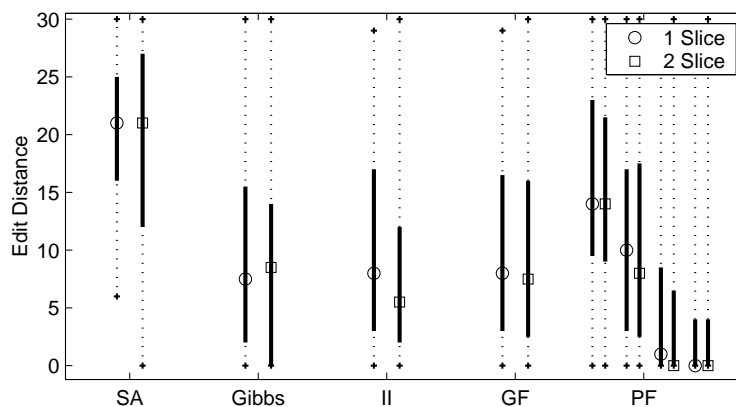


Figure 4.6: Typical runs of Gibbs sampling, Simulated Annealing (SA) and Iterative Improvement (II) on clavichord example. All algorithms are initialized to the greedy filter solution. The annealing schedule for SA was linear from $\rho_1 = 0.1$ to $\rho_{33} = 10$ and then proceeding deterministically by $\rho_{34:50} = \infty$. When SA or II converge to a configuration, we reinitialize by a particle filter with one particle that draws randomly proportional to the optimal proposal. Sharp drops in the likelihood correspond to reinitializations. We see that, at the first sweep, the greedy filter solution can only be slightly improved by II. Consequently the sampler reinitializes. The likelihood of SA drops considerably, mainly due to the high temperature, and consequently stabilizes at a suboptimal solution. The Gibbs sampler seems to explore the support of the posterior but is not able to visit the MAP state in this run.



(a) Likelihood Difference



(b) Edit Distance. MCMC results with 10 sweeps are omitted.

Figure 4.7: Comparison of inference methods on the clave data. The squares and ovals denote the median and the vertical bars correspond to the interval between %25 and %75 quantiles. We have tested the MCMC methods (Gibbs, SA and II) independently for 10 and 50 (shown from left to right). The SMC methods are the greedy filter (GF) and particle filter (PF). We have tested filters with $N = \{5, 10, 50, 100\}$ particles independently (shown from left to right.).

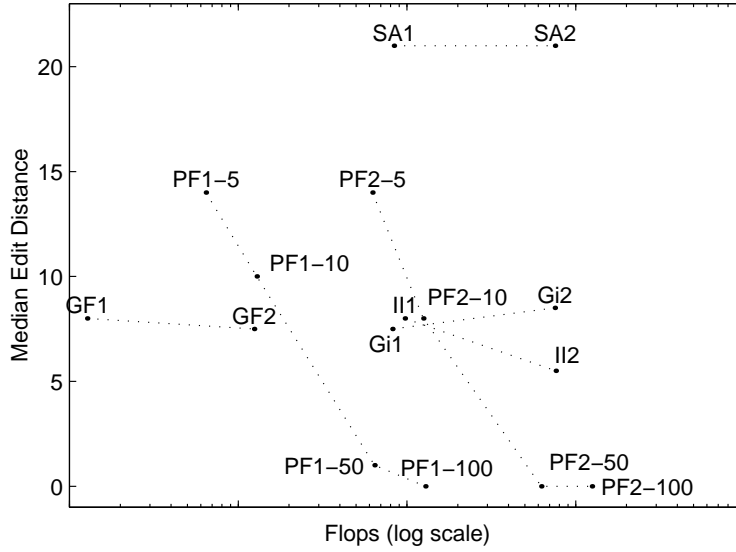


Figure 4.8: Comparison of execution time in terms of floating point operations. For all methods, the first number (1 or 2) denotes the number slices used by the optimal proposal distribution. For the particle filter (PF), the second number denotes the number of particles. The dashed lines are merely used to connect related methods.

realistic range and all according to the judgment of the performer). Further details are reported in (Cemgil et al., 2001).

Preprocessing

The original performances contained several errors, such as missing notes or additional notes that were not on the original score. Such errors are eliminated by using a matching technique (Heijink et al., 2000) based on dynamical programming. However, visual inspection of the resulting dataset suggested still several matching errors that we interpret as outliers. To remove these outliers, we have extended the quantization model with a two state switching observation model, i.e., the discrete space consists of (γ_k, i_k) . In this simple outlier detection mechanism, each switch i_k is a binary indicator variable specifying whether the onset y_k is an outlier or not. We assume that all indicators are independent a-priori and have a uniform prior. The observation model is given by $p(y_k | i_k, \tau_k) = \mathcal{N}(0, R_{i_k})$ ⁵. Since the score $\gamma_{1:K}$ is known, the only unknown discrete quantities are the indicators $i_{0:K}$. We have used greedy filtering followed by iterative improvement to find the MAP state of indicators $i_{0:K}$ and eliminated outliers in our further studies. For many performances, there were around 2 – 4 outliers, less than 1% of all the notes. The resulting dataset can be downloaded from the url <http://www.snn.kun.nl/~cemgil>.

Parameter Estimation

We have trained tempo tracking models with different dimensionality D , where D denotes the dimension of the hidden variable z . In all of the models, we use a transition matrix that has the form in Eq. 4.8.

Since the true score is known, i.e., the score position c_k of each onset y_k is given, we can

⁵We took $R_{i_k=0} = 0.002$ and $R_{i_k=1} = 2$.

clamp all the discrete variables in the model. Consequently, we can estimate the observation noise variance R , the transition noise variance Q and the transition matrix coefficients A from data.

We have optimized the parameters by Expectation-Maximization (EM) for the linear dynamical systems (Shumway & Stoffer, 1982; Ghahramani & Hinton, 1996) using all performances of “Yesterday” as training data. Similarly, the score prior parameters are estimated by frequency counts from the score of “Yesterday”⁶. All tests are carried out on “Michelle”.

Results

In Figure 4.9 we show the result of typesetting a performance with and without tempo tracking. Due to fluctuations in tempo, the quality of the automatically generated score is very poor. The quality can be significantly improved by using our model.

Figure 4.10 shows some tempo tracking examples on Michelle dataset for pianists from different background and training. We observe that in most cases the results are satisfactory.

In Figure 4.11, we give a summary of test results on Michelle data in terms of the loglikelihood and edit distance as a function of model order and number of particles used for inference. Figure 4.11(a) shows that the median likelihood on test data is increasing with model order. This suggests that a higher order filter is able to capture structure in pianists’ expressive timing. Moreover, as for the sythetic data, we see a somewhat monotonic increase in the likelihood of solutions found when using more particles.

The edit distance between the original score and the estimates are given in Figure 4.11(b). Since both pieces are arranged for piano, due to polyphony, there are many onsets that are associated with the same score position. Consequently, many γ_k^{true} in the original score are effectively zero. In such cases, typically, the corresponding inter onset interval $y_k - y_{k-1}$ is also very small and the correct quantization (namely $\gamma_k = 0$) can be identified even if the tempo estimate is completely wrong. As a consequence, the edit distance remains small. To make the task slightly more challenging, we exclude the onsets with $\gamma_k^{\text{true}} = 0$ from edit distance calculation.

We observe that the extra prediction ability obtained using a higher order model does not directly translate to a better transcription. The errors are around 5% for all models. On the other hand, the variance of edit distance for higher order models is smaller suggesting an increased robustness towards divergence from the tempo track implied by the original score.

4.5 Discussion

We have presented a switching state space model for joint rhythm quantization and tempo tracking. The model describes the rhythmic structure of musical pieces by a prior distribution over score positions. In this representation, it is easy to construct a generic prior that prefers simpler notations and to learn parameters from a data set. The prior on score positions $c_{0:K}$ translates to a non-Markovian distribution over a score $\gamma_{1:K}$.

Timing deviations introduced by performers (tempo fluctuation, accentuations and motor errors) are modeled as independent Gaussian noise sources. Performer specific timing preferences are captured by the parameters of these distributions.

Given the model, we have formulated rhythm quantization as a MAP state estimation problem and tempo tracking as a filtering problem. We have introduced Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) to approximate the respective distributions.

⁶The maximum likelihood parameters for a model of dimension $D = 3$ are found to be: $a = -0.072$, $R = 0.013^2$ and $q_\tau = 0.008^2$, $q_{\Delta_1} = 0.007^2$ and $q_{\Delta_2} = 0.050^2$. The prior $p(c)$ is $p(0) = 0.80$, $p(1/3) = 0.0082$, $p(1/2) = 0.15$ $p(5/6) = 0.0418$. Remaining $p(c)$ are set to 10^{-6} .

Michelle
Lennon/McCartney

(a) Original Score
(b) Typesetting without processing by the model. Due to fluctuations in tempo, the quality of the score is poor.
(c) Typesetting after tempo tracking and quantization with a particle filter.

Figure 4.9: Results of Typesetting the scores.

The quantization model we propose is similar to that of (Raphael, 2001a). For transcription, Raphael proposes to compute $\arg \max p(c_{0:K}, z_{0:K} | y_{0:K})$ and uses a message propagation scheme that is essentially analogous to Rao-Blackwellized particle filtering. To prevent the number of kernels from explosion, he uses a deterministic selection method, called “thinning”. The advantage of Raphael’s approach is that the joint MAP trajectory can be computed exactly, provided that the continuous hidden state z is one dimensional and the model is in a parameter regime that keeps the number of propagated Gaussian kernels limited, e.g., if R is small, thinning can not eliminate many kernels. One disadvantage is that the number of kernels varies depending upon the features of the filtering distribution; it is difficult to implement such a scheme in real time. Perhaps more importantly, simple extensions such as increasing the dimensionality of z or introducing nonlinearities to the transition model would render the approach quickly invalid. In contrast, Monte Carlo methods provide a generic inference technique that allow great flexibility in models one can employ.

We have tested our method on a challenging artificial problem (clave example). SMC has outperformed MCMC in terms of the quality of solutions, as measured in terms of the likelihood as well as the edit distance. We propose the use of SMC for both problems. For finding the MAP quantization, we propose to apply iterative improvement (II) to the SMC solution on the reduced configuration space.

The correct choice of the score prior is important in the overall performance of the system. Most music pieces tend to have a certain rhythmical vocabulary, that is certain rhythmical motives reoccur several times in a given piece. The rhythmic structure depends mostly upon the musical genre and composer. It seems to be rather difficult to devise a general prior model that would work well in a large spectrum of styles. Nevertheless, for a given genre, we expect a simple prior to capture enough structure sufficient for good transcription. For example, for the Beatles dataset, we have estimated the prior by counting from the original score of “Yesterday”. The statistics are

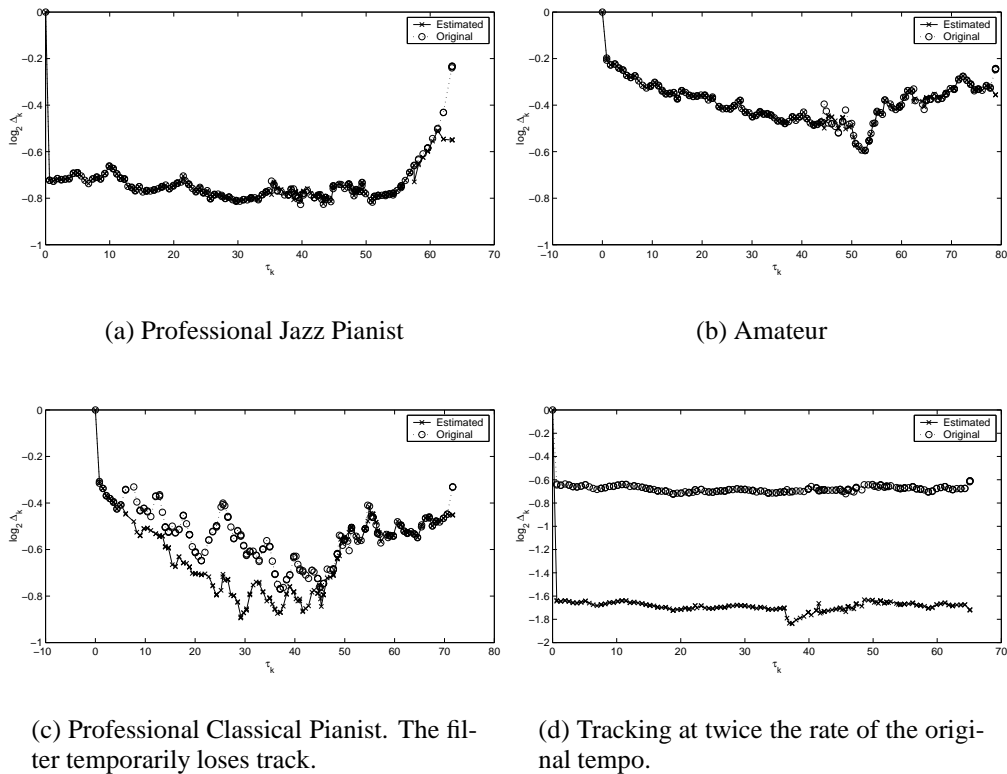
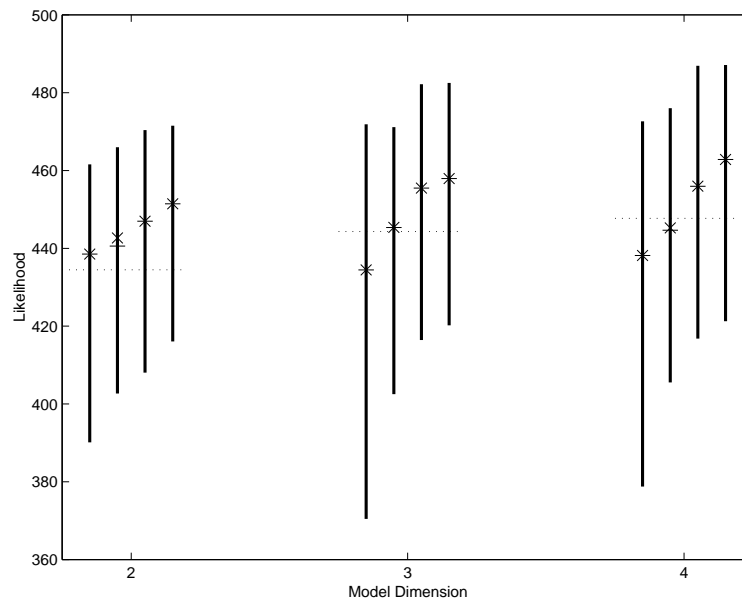
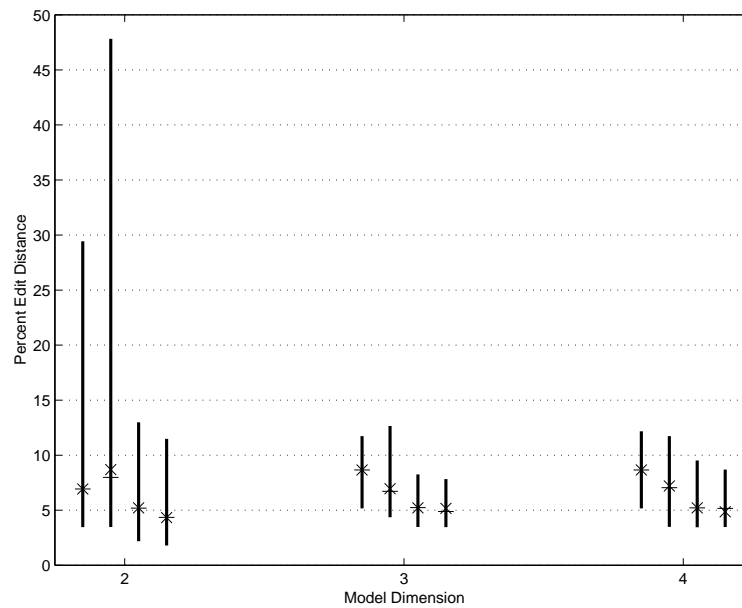


Figure 4.10: Examples of filtered estimates of $z_{0:K} = [\tau_k, \Delta_k]^T$ from the Beatles data set. Circles denote the mean of $p(z_k | \gamma_{1:k}^{\text{original}}, y_{0:k})$ and “x” denote mean $p(z_k | \gamma_{1:k}^*, y_{0:k})$ obtained by SMC. It is interesting to note different timing characteristics. For example the classical pianist uses a lot more tempo fluctuation than the professional jazz pianist. Jazz pianist slows down dramatically at the end of the piece, the amateur “rushes”, i.e., constantly accelerates at the beginning. The tracking and quantization results for (a) and (b) are satisfactory. In (a), the filter loses track at the last two notes, where the pianist dramatically slows down. In (c), the filter loses track but catches up again. In (d), the filter jumps to a metrical level that is twice as fast as the original performance. That would translate to a duplication in note durations only.



(a) Likelihood. The dashed horizontal line shows the median likelihood of the original score of Michelle under each model.



(b) Edit Distance

Figure 4.11: SMC results on the test data (108 performances of Michelle). For each model we show the results obtained with $N = 1, 10, 20$ and 50 particles. The “-” show the median of the best particle and “x” denote the median after applying iterative improvement. The vertical bars correspond to the interval between %25 and %75 quantiles.

fairly close to that of “Michelle”. The good results on the test set can be partially accounted for the fact that both pieces have a similar rhythmical structure.

Conditioned on the score, the tempo tracking model is a linear dynamical system. We have optimized several tempo models using EM where we have varied the dimension of tempo variables z . The test results suggest that increasing the dimensionality of z improves the likelihood. However, increase in the likelihood of the whole dataset does not translate directly to overall better quantization results (as measured by edit distance). We observe that models trained on the whole training data fail consistently for some subjects, especially professional classical pianists. Perhaps interestingly, if we train “custom” models specifically optimized for the same subjects, we can improve results significantly also on test cases. This observation suggests a kind of multimodality in the parameter space where modes correspond to different performer regimes. It seems that a Kalman filter is able to capture the structure in expressive timing deviations. However, when averaged over all subjects, these details tend to be wiped out, as suggested by the quantization results that do not vary significantly among models of different dimensions.

A related problem with the edit distance measure is that under an “average” model, the likelihood of the desired score (e.g., original score of “Michelle”) may have a lower likelihood than a solution found by an inference method. In such cases increasing the likelihood may even decrease the edit distance. In some test cases we even observe solutions with a higher likelihood than the original notation where all notes are wrong. In most of these cases, the tempo trajectory of the solution correspond to the half or twice of the original tempo so consequently all note durations are halved or doubled (e.g., all whole notes are notated as half notes, all half notes as quarters e.t.c.). Considering the fact that the model is “self initializing” its tempo, that is we assume a broad uncertainty a-priori, the results are still satisfactory from a practical application perspective.

One potential shortcoming of our model is that it takes only timing information of onsets into account. In reality, we believe that pitch and melodic grouping as well as articulation (duration between note onsets and offsets) and dynamics (louder or softer) provide useful additional information for tempo tracking as well as quantization. Moreover, current model assumes that all onsets are equally relevant for estimation. That is probably in general not true: for example, a kick-drum should provide more information about the tempo than a flute. On the other hand, our simulations suggest that even from such a limited model one can obtain quite satisfactory results, at least for simple piano music.

It is somewhat surprising, that SMC, basically a method that samples from the filtering distribution outperforms an MCMC method such as SA that is specifically designed for finding the MAP solution given all observations. An intuitive explanation for relatively poorer MCMC results is that MCMC proceeds first by proposing a global solution and then tries to improve it by local adjustments. A human transcriber, on the other hand, would listen to shorter segments of music and gradually write down the score. In that respect, the sequential update schema of SMC seems to be more natural for the rhythm transcription problem. Similar results, where SMC outperforms MCMC are already reported in the literature, e.g., in the so-called “Growth Monte Carlo” for generating self-avoiding random walks (Liu, Chen, & Logvinenko, 2001). It seems that for a large class of dynamical problems, including rhythm transcription, sequential updating is preferable over batch methods.

We note that theoretical convergence results for SA require the use of a logarithmic cooling schedule. It seems that our cooling schedule was too fast to meet this requirement; so one has to be still careful in interpreting the poor performance as a negative SA result. We maintain that by using a richer neighborhood structure in the configuration space (e.g., by using a block proposal distribution) and a slower cooling schedule, SA results can be improved significantly. Moreover, MCMC methods can be also be modified to operate sequentially, for example see (Marthi, Pasula, Russell, & Peres, 2002).

Another family of inference methods for switching state space models rely on deterministic

approximate methods. This family includes variational approximations (Ghahramani & Hinton, 1998) and expectation propagation (Heskes, 2002). It remains an interesting open question whether deterministic approximation methods provide an advantage in terms of computation time and accuracy; in particular for the quantization problem and for other switching state space models. A potential application of the deterministic approximation techniques in a MCMC schema can be in designing proposal distributions that extend over several time slices. Such a schema would circumvent the burden for computing the optimal proposal distribution exhaustively hence allowing more global moves for the sampler.

Our current results suggest the superiority of SMC for our problem. Perhaps the most important advantage of SMC is that it is essentially an “anytime” algorithm; if we have a faster computer we can increase the number of particles to make use of the additional computational power. When computing time becomes short one can decrease the number of samples. These features make SMC very attractive for real-time applications where one can easily tune the quality/computation-time tradeoff.

Motivated by the practical advantages of SMC and our positive simulation results, we have implemented a prototype of SMC method in real-time. Our current computer system (a 800 MHz P3 laptop PC running MS Windows) allows us to use up to 5 particles with almost no delay even during busy passages. We expect to significantly improve the efficiency by translating the MATLAB[®] constructs to native C code. Hence, the method can be used as a tempo tracker in an automatic interactive performance system and as a quantizer in an automatic score typesetting program.

Appendix 4.A A generic prior model for score positions

In traditional western music notation, note durations are generated by recursive subdivisions starting from a whole note, hence it is also convenient to generate score positions in a similar fashion by regular subdivisions. We decompose a score position into an integer part and a fraction: $c = \lfloor c \rfloor + (c \bmod 1)$. For defining a prior, we will only use the fraction.

The set of all fractions can be generated by recursively subdividing the unit interval $[0, 1)$. We let $\mathcal{S} = [s_i]$ denote a subdivision schema, where $[s_i]$ is a (finite) sequence of arbitrary integers (usually small primes such as 2,3 or 5). The choice of a particular \mathcal{S} depends mainly on the assumed time signature. We generate the set of fractions C as follows: At first iteration, we divide the unit interval into s_1 intervals of equal length and append the endpoints c' of resulting intervals into the set C . At each following iteration i , we subdivide all intervals generated by the previous iteration into s_i equal parts and append all resulting endpoints to C . Note that this procedure generates a regular grid where two neighboring grid points have the distance $1/\prod_i s_i$. We denote the iteration number at which the endpoint c' is first inserted to C as the *depth* of c' (with respect to \mathcal{S}). This number will be denoted as $d(c'|\mathcal{S})$. It is easy to see that this definition of d coincides with the number of significant bits to represent $c \bmod 1$ when $\mathcal{S} = [2, 2, \dots]$.

As an illustrative example consider the subdivision $\mathcal{S} = [3, 2]$. At the first iteration, the unit interval is divided into $s_1 = 3$ equal intervals, and the resulting endpoints 0, 1/3, and 2/3 are inserted into C with depths $d(0) = d(1/3) = d(2/3) = 1$. At the second iteration, the new endpoints 1/6, 3/6 and 5/6 are inserted to C and are assigned the depth 2.

Given an \mathcal{S} , we can define a distribution on score positions

$$p(c_k|\mathcal{S}) \propto \exp(-\lambda d(c_k \bmod 1|\mathcal{S}))$$

If we wish to consider several time signatures, i.e., different subdivision schemata, we can interpret \mathcal{S} as a hidden indicator variable and define a prior $p(\mathcal{S})$. In this case, the prior becomes a multinomial mixture given by $p(c_k) = \sum_{\mathcal{S}} p(c_k|\mathcal{S})p(\mathcal{S})$. For further details and empirical results justifying such a choice see (Cemgil et al., 2000).

Appendix 4.B Derivation of two pass Kalman filtering Equations

Consider a Gaussian potential with mean μ and covariance Σ defined on some domain indexed by x .

$$\phi(x) = Z \times \mathcal{N}(\mu, \Sigma) = Z |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (4.28)$$

where $\int dx \phi(x) = Z > 0$. If $Z = 1$ the potential is normalized. The exponent in Eq. 4.28 is a quadratic form so the potential can be written as

$$\phi(x) = \exp\left(g + h^T x - \frac{1}{2} x^T K x\right) \quad (4.29)$$

where

$$K = \Sigma^{-1} \quad h = \Sigma^{-1} \mu \quad g = \log Z + \frac{1}{2} \log \left| \frac{K}{2\pi} \right| - \frac{1}{2} h^T K^{-1} h$$

To denote a potential in canonical form we will use the notation

$$\phi(x) = Z \times \mathcal{N}(\mu, \Sigma) \equiv [h, K, g]$$

and we will refer to g , h and K as *canonical* parameters. Now we consider a Gaussian potential on $(x_1, x_2)^T$. The canonical representation is

$$\phi(x_1, x_2) = \left[\begin{pmatrix} h_1 \\ h_2 \end{pmatrix}, \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}, g \right]$$

In models where several variables are interacting, one can find desired quantities by applying three basic operations defined on Gaussian potentials. Those are *multiplication*, *conditioning*, and *marginalization*. The multiplication of two Gaussian potentials on the same index set x follows directly from Eq. 4.29 and is given by

$$\begin{aligned} \phi'(x) &= \phi_a(x) \times \phi_b(x) \\ [h', K', g'] &= [h_a, K_a, g_a] \times [h_b, K_b, g_b] = [h_a + h_b, K_a + K_b, g_a + g_b] \end{aligned}$$

If the domain of ϕ_a and ϕ_b only overlaps on a subset, then potentials are extended to the appropriate domain by appending zeros to the corresponding dimensions.

The marginalization operation is given by

$$\phi(x_1) = \int_{x_2} \phi(x_1, x_2) = [h_1 - K_{12} K_{22}^{-1} h_2, K_{11} - K_{12} K_{22}^{-1} K_{21}, g']$$

where $g' = g - \frac{1}{2} \log |K_{22}/2\pi| + \frac{1}{2} h_2^T (K_{22})^{-1} h_2$ and g is the initial constant term of $\phi(x_1, x_2)$. The conditioning operation is given by

$$\phi(x_1, x_2 = \hat{x}_2) = [h_1 - K_{12} \hat{x}_2, K_{11}, g']$$

where $g' = g + h_2^T \hat{x}_2 - \frac{1}{2} \hat{x}_2^T K_{22} \hat{x}_2$.

4.B.1 The Kalman Filter Recursions

Suppose we are given the following linear model subject to noise

$$\begin{aligned} z_k &= Az_{k-1} + \zeta_k \\ y_k &= Cz_k + \epsilon_k \end{aligned}$$

where A and C are constant matrices, $\zeta_k \sim \mathcal{N}(0, Q)$ and $\epsilon_k \sim \mathcal{N}(0, R)$
The model encodes the joint distribution

$$p(z_{1:K}, y_{1:K}) = \prod_{k=1}^K p(y_k|z_k)p(z_k|z_{k-1}) \quad (4.30)$$

$$p(z_1|z_0) = p(z_1) \quad (4.31)$$

$$\begin{aligned} p(z_1) &= [P^{-1}\mu, P^{-1}, -\frac{1}{2}\log|2\pi P| - \frac{1}{2}\mu^T P^{-1}\mu] \\ p(y_1|z_1) &= \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} C^T R^{-1} C & -C^T R^{-1} \\ -R^{-1} C & R^{-1} \end{pmatrix}, -\frac{1}{2}\log|2\pi R| \right] \\ p(y_1 = \hat{y}_1|z_1) &= [0 + C^T R^{-1} \hat{y}_1, C^T R^{-1} C, -\frac{1}{2}\log|2\pi R| - \frac{1}{2}\hat{y}_1^T R^{-1} \hat{y}_1] \\ p(z_2|z_1) &= \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} A^T Q^{-1} A & -A^T Q^{-1} \\ -Q^{-1} A & Q^{-1} \end{pmatrix}, -\frac{1}{2}\log|2\pi Q| \right] \\ \dots & \end{aligned}$$

Forward Message Passing

Suppose we wish to compute the likelihood

$$p(y_{1:K}) = \int_{z_K} p(y_K|z_K) \dots \int_{z_2} p(z_3|z_2)p(y_2|z_2) \int_{z_1} p(z_2|z_1)p(y_1|z_1)p(z_1)$$

⁷We can compute this integral by starting from z_1 and proceeding to z_K . We define forward “messages” α as

- $\alpha_{1|0} = p(z_1)$
- $k = 1 : K$
 - $\alpha_{k|k} = p(y_k = \hat{y}_k|z_k)\alpha_{k|k-1}$
 - $\alpha_{k+1|k} = \int_{z_k} p(z_{k+1}|z_k)\alpha_{k|k}$

The forward recursion is given by

- $\alpha_{1|0} = [P^{-1}\mu, P^{-1}, -\frac{1}{2}\log|2\pi P| - \frac{1}{2}\mu^T P^{-1}\mu]$
- $k = 1 \dots K$
 - $\alpha_{k|k} = [h_{k|k}, K_{k|k}, g_{k|k}]$

⁷We let $\int_z \equiv \int dz$

$$\begin{aligned}
h_{k|k} &= C^T R^{-1} \hat{y}_k + h_{k|k-1} \\
K_{k|k} &= C^T R^{-1} C + K_{k|k-1} \\
g_{k|k} &= g_{k|k-1} - \frac{1}{2} \log |2\pi R| - \frac{1}{2} \hat{y}_k^T R^{-1} \hat{y}_k \\
- \alpha_{k+1|k} &= [h_{k+1|k}, K_{k+1|k}, g_{k+1|k}] \\
M_k &= (A^T Q^{-1} A + K_{k|k})^{-1} \\
h_{k+1|k} &= Q^{-1} A M_k h_{k|k} \\
K_{k+1|k} &= Q^{-1} - Q^{-1} A M_k A^T Q^{-1} \\
g_{k+1|k} &= g_{k|k} - \frac{1}{2} \log |2\pi Q| + \frac{1}{2} \log |2\pi M_k| + \frac{1}{2} h_{k|k}^T M_k h_{k|k}
\end{aligned}$$

Backward Message Passing

We can compute the likelihood also by starting from y_K .

$$p(y_{1:K}) = \int_{z_1} p(z_1) p(y_1|z_1) \int_{z_2} p(z_2|z_1) p(y_2|z_2) \dots \int_{z_K} p(z_K|z_{K-1}) p(y_K|z_K)$$

In this case the backward propagation can be summarized as

- $\beta_{K|K+1} = 1$
- $k = K \dots 1$
 - $\beta_{k|k} = p(y_k = \hat{y}_k | z_k) \beta_{k|k+1}$
 - $\beta_{k-1|k} = \int_{z_k} p(z_k | z_{k-1}) \beta_{k|k}$

The recursion is given by

- $[h_{K|K+1}^*, K_{K|K+1}^*, g_{K|K+1}^*] = [0, 0, 0]$
- $k = K \dots 1$
 - $\beta_{k|k} = [h_{k|k}^*, K_{k|k}^*, g_{k|k}^*]$

$$\begin{aligned}
h_{k|k}^* &= C^T R^{-1} \hat{y}_k + h_{k|k+1}^* \\
K_{k|k}^* &= C^T R^{-1} C + K_{k|k+1}^* \\
g_{k|k}^* &= -\frac{1}{2} \log |2\pi R| - \frac{1}{2} \hat{y}_k^T R^{-1} \hat{y}_k + g_{k|k+1}^*
\end{aligned}$$
 - $\beta_{k-1|k} = [h_{k-1|k}^*, K_{k-1|k}^*, g_{k-1|k}^*]$

$$\begin{aligned}
M_k^* &= (Q^{-1} + K_{k|k}^*)^{-1} \\
h_{k-1|k}^* &= A^T Q^{-1} M_k^* h_{k|k}^* \\
K_{k-1|k}^* &= A^T Q^{-1} (Q - M_k^*) Q^{-1} A \\
g_{k-1|k}^* &= g_{k|k}^* - \frac{1}{2} \log |2\pi Q| + \frac{1}{2} \log |2\pi M_k^*| + \frac{1}{2} h_{k|k}^{*T} M_k^* h_{k|k}^*
\end{aligned}$$

Kalman Smoothing

Suppose we wish to find the distribution of a particular z_k given all the observations $y_{1:K}$. We just have to combine forward and backward messages as

$$\begin{aligned}
p(z_k | y_{1:K}) &\propto p(y_{k+1:K}, z_k, y_{1:k}) \\
&= p(y_{1:k}, z_k) p(y_{k+1:K} | z_k) \\
&= \alpha_{k|k} \times \beta_{k|k+1} \\
&= [h_{k|k} + h_{k|k+1}^*, K_{k|k} + K_{k|k+1}^*, g_{k|k} + g_{k|k+1}^*]
\end{aligned}$$

Appendix 4.C Rao-Blackwellized SMC for the Switching State space Model

We let $i = 1 \dots N$ be an index over particles and $s = 1 \dots S$ an index over states of γ . We denote the (unnormalized) filtering distribution at time $k - 1$ by

$$\phi_{k-1}^{(i)} \hat{=} p(y_{0:k-1}, z_{k-1} | \gamma_{1:k-1}^{(i)})$$

Since $y_{0:k-1}$ are observed, $\phi_{k-1}^{(i)}$ is a Gaussian potential on z_{k-1} with parameters $Z_{k-1}^{(i)} \times \mathcal{N}(\mu_{k-1}^{(i)}, \Sigma_{k-1}^{(i)})$.

Note that the normalization constant $Z_{k-1}^{(i)}$ is the data likelihood $p(y_{0:k-1} | \gamma_{1:k-1}^{(i)}) = \int dz_k \phi_{k-1}^{(i)}$. Similarly, we denote the filtered distribution at the next slice conditioned on $\gamma_k = s$ by

$$\begin{aligned} \phi_k^{(s|i)} &\hat{=} \int dz_{k-1} p(y_k | z_k) p(z_k | z_{k-1}, \gamma_k = s) \phi_{k-1}^{(i)} \\ &= p(y_{0:k}, z_k | \gamma_{1:k-1}^{(i)}, \gamma_k = s) \end{aligned} \quad (4.32)$$

We denote the normalization constant of $\phi_k^{(s|i)}$ by $Z_k^{(s|i)}$. Hence the joint proposal on s and (i) is given by

$$\begin{aligned} q_k^{(s|i)} &= \int dz_k \phi_k^{(s|i)} \times p(\gamma_k = s, \gamma_{1:k-1}^{(i)}) \\ &= p(\gamma_k = s, \gamma_{1:k-1}^{(i)}, y_{0:k}) \end{aligned}$$

The outline of the algorithm is given below:

- Initialize. For $i = 1 \dots N$, $\phi_0^{(i)} \leftarrow p(y_0, x_0)$
- For $k = 1 \dots K$
 - For $i = 1 \dots N$, $s = 1 \dots S$
 - Compute $\phi_k^{(s|i)}$ from $\phi_{k-1}^{(i)}$ using Eq.4.32.
 - $q_k^{(s|i)} \leftarrow Z_k^{(s|i)} \times p(\gamma_k = s, \gamma_{1:k-1}^{(i)})$
 - For $i = 1 \dots N$
 - Select a tuple $(s|j) \sim q_k$
 - $\gamma_{1:k}^{(i)} \leftarrow (\gamma_{1:k-1}^{(j)}, \gamma_k = s)$
 - $\phi_k^{(i)} \leftarrow \phi_k^{(s|j)}$
 - $w_k^{(i)} \leftarrow \sum_s q_k^{(s|j)}$

Note that the procedure has a “built-in” resampling schema for eliminating particles with small importance weight. Sampling jointly on $(s|i)$ is equivalent to sampling a single s for each i and then resampling i according to the weights $w_k^{(i)}$. One can also check that, since we are using the optimal proposal distribution of Eq.4.27, the weight at each step is given by $w_k^{(i)} = p(\gamma_{1:k-1}^{(i)}, y_{0:k})$.

Chapter 5

Piano-Roll Inference

In this paper we present a graphical model for polyphonic music transcription. Our model, formulated as a Dynamical Bayesian Network, embodies a transparent and computationally tractable approach to this acoustic analysis problem. An advantage of our approach is that it places emphasis on explicitly modelling the sound generation procedure. It provides a clear framework in which both high level (cognitive) prior information on music structure can be coupled with low level (acoustic physical) information in a principled manner to perform the analysis. The model is a special case of the, generally intractable, switching Kalman filter model. Where possible, we derive, exact polynomial time inference procedures, and otherwise efficient approximations. We argue that our generative model based approach is computationally feasible for many music applications and is readily extensible to more general auditory scene analysis scenarios.

Adapted from A. T. Cemgil, H. J. Kappen, and D. Barber. *A generative model for music transcription*. Accepted to IEEE Transactions on Speech and Audio Processing, 2004.

5.1 Introduction

When humans listen to sound, they are able to associate acoustical signals generated by different mechanisms with individual symbolic events (Bregman, 1990). The study and computational modelling of this human ability forms the focus of computational auditory scene analysis (CASA) and machine listening (Brown & Cooke, 1994). Research in this area seeks solutions to a broad range of problems such as the cocktail party problem, (for example automatically separating voices of two or more simultaneously speaking persons, see e.g. (Weintraub, 1985; Roweis, 2001)), identification of environmental sound objects (Ellis, 1996) and musical scene analysis (Scheirer, 2000). Traditionally, the focus of most research activities has been in speech applications. Recently, analysis of musical scenes is drawing increasingly more attention, primarily because of the need for content based retrieval in very large digital audio databases (Tzanetakis, 2002) and increasing interest in interactive music performance systems (Rowe, 2001).

5.1.1 Music Transcription

One of the hard problems in musical scene analysis is automatic music transcription, that is, the extraction of a human readable and interpretable description from a recording of a music performance. Ultimately, we wish to infer automatically a musical notation (such as the traditional western music notation) listing the pitch levels of notes and corresponding time-stamps for a given performance. Such a representation of the surface structure of music would be very useful in a

broad spectrum of applications such as interactive music performance systems, music information retrieval (Music-IR) and content description of musical material in large audio databases, as well as in the analysis of performances. In its most unconstrained form, i.e., when operating on an arbitrary polyphonic acoustical input possibly containing an unknown number of different instruments, automatic music transcription remains a great challenge. Our aim in this paper is to consider a computational framework to move us closer to a practical solution of this problem.

Music transcription has attracted significant research effort in the past – see (Scheirer, 2000) and (Plumbley et al., 2002) for a detailed review of early and more recent work, respectively. In speech processing, the related task of tracking the pitch of a single speaker is a fundamental problem and methods proposed in the literature are well studied (Hess, 1983). However, most current pitch detection algorithms are based largely on heuristics (e.g., picking high energy peaks of a spectrogram, correlogram, auditory filter bank, etc.) and their formulation usually lacks an explicit objective function or signal model. It is often difficult to theoretically justify the merits and shortcomings of such algorithms, and compare them objectively to alternatives or extend them to more complex scenarios.

Pitch tracking is inherently related to the detection and estimation of sinusoids. The estimation and tracking of single or multiple sinusoids is a fundamental problem in many branches of applied sciences, so it is less surprising that the topic has also been deeply investigated in statistics, (e.g. see (Quinn & Hannan, 2001)). However, ideas from statistics seem to be not widely applied in the context of musical sound analysis, with only a few exceptions (Irizarry, 2001, 2002) who present frequentist techniques for very detailed analysis of musical sounds with particular focus on decomposition of periodic and transient components. Saul et al. (2002) has presented real-time monophonic pitch tracking application based on a Laplace approximation to the posterior parameter distribution of an AR(2) model (Truong-Van, 1990; Quinn & Hannan, 2001, page 19). Their method outperforms several standard pitch tracking algorithms for speech, suggesting potential practical benefits of an approximate Bayesian treatment. For monophonic speech, a Kalman filter based pitch tracker is proposed by Parra and Jain (2001) that tracks parameters of a harmonic plus noise model (HNM). They propose the use of Laplace approximation around the predicted mean instead of the extended Kalman filter (EKF). For both methods, however, it is not obvious how to extend them to polyphony.

Kashino Kashino et al. (1995) is, to our knowledge, the first author to apply graphical models explicitly to the problem of polyphonic music transcription. Sterian Sterian (1999) described a system that viewed transcription as a model driven segmentation of a time-frequency image. Walmsley Walmsley (2000) treats transcription and source separation in a full Bayesian framework. He employs a frame based generalized linear model (a sinusoidal model) and proposes inference by reversible-jump Markov Chain Monte Carlo (MCMC) algorithm. The main advantage of the model is that it makes no strong assumptions about the signal generation mechanism, and views the number of sources as well as the number of harmonics as unknown model parameters. Davy and Godsill Davy and Godsill (2003) address some of the shortcomings of his model and allow changing amplitudes and frequency deviations. The reported results are encouraging, although the method is computationally very expensive.

5.1.2 Approach

Musical signals have a very rich temporal structure, both on a physical (signal) and a cognitive (symbolic) level. From a statistical modelling point of view, such a hierarchical structure induces very long range correlations that are difficult to capture with conventional signal models. Moreover, in many music applications, such as transcription or score following, we are usually interested in a symbolic representation (such as a score) and not so much in the “details” of the actual waveform. To abstract away from the signal details, we define a set of intermediate variables (a

sequence of indicators), somewhat analogous to a “piano-roll” representation. This intermediate layer forms the “interface” between a symbolic process and the actual signal process. Roughly, the symbolic process describes how a piece is composed and performed. We view this process as a prior distribution on the piano-roll. Conditioned on the piano-roll, the signal process describes how the actual waveform is synthesized.

Most authors view automated music transcription as an “audio to piano-roll” conversion and usually consider “piano-roll to score” a separate problem. This view is partially justified, since source separation and transcription from a polyphonic source is already a challenging task. On the other hand, automated generation of a human readable score includes nontrivial tasks such as tempo tracking, rhythm quantization, meter and key induction (Raphael, 2001a; Temperley, 2001; Cemgil & Kappen, 2003). As also noted by other authors (e.g. (Kashino et al., 1995; Martin, 1999; Klapuri, Virtanen, & Holm, 2000)), we believe that a model that integrates this higher level symbolic prior knowledge can guide and potentially improve the inferences, both in terms quality of a solution and computation time.

There are many different natural generative models for piano-rolls. In (Cemgil et al., 2003), we proposed a realistic hierarchical prior model. In this paper, we consider computationally simpler prior models and focus more on developing efficient inference techniques of a piano-roll representation. The organization of the paper is as follows: We will first present a generative model, inspired by additive synthesis, that describes the signal generation procedure. In the sequel, we will formulate two subproblems related to music transcription: melody identification and chord identification. We will show that both problems can be easily formulated as combinatorial optimization problems in the framework of our model, merely by redefining the prior on piano-rolls. Under our model assumptions, melody identification can be solved exactly in polynomial time (in the number of samples). By deterministic pruning, we obtain a practical approximation that works in linear time. Chord identification suffers from combinatorial explosion. For this case, we propose a greedy search algorithm based on iterative improvement. Consequently, we combine both algorithms for polyphonic music transcription. Finally, we demonstrate how (hyper-)parameters of the signal process can be estimated from real data.

5.2 Polyphonic Model

In a statistical sense, music transcription, (as many other perceptual tasks such as visual object recognition or robot localization) can be viewed as a latent state estimation problem: given the audio signal, we wish to identify the sequence of events (e.g. notes) that gave rise to the observed audio signal.

This problem can be conveniently described in a Bayesian framework: given the audio samples, we wish to infer a piano-roll that represents the onset times (e.g. times at which a ‘string’ is ‘plucked’), note durations and the pitch classes of individual notes. We assume that we have one microphone, so that at each time t we have a one dimensional observed quantity y_t . Multiple microphones (such as required for processing stereo recordings) would be straightforward to include in our model. We denote the temporal sequence of audio samples $\{y_1, y_2, \dots, y_t, \dots, y_T\}$ by the shorthand notation $y_{1:T}$. A constant sampling frequency F_s is assumed.

Our approach considers the quantities we wish to infer as a collection of ‘hidden’ variables, whilst acoustic recording values $y_{1:T}$ are ‘visible’ (observed). For each observed sample y_t , we wish to associate a higher, unobserved quantity that labels the sample y_t appropriately. Let us denote the unobserved quantities by $\mathcal{H}_{1:T}$ where each \mathcal{H}_t is a vector. Our hidden variables will contain, in addition to a piano-roll, other variables required to complete the sound generation

procedure. We will elucidate their meaning later. As a general inference problem, the posterior distribution is given by Bayes' rule

$$p(\mathcal{H}_{1:T}|y_{1:T}) \propto p(y_{1:T}|\mathcal{H}_{1:T})p(\mathcal{H}_{1:T}) \quad (5.1)$$

The likelihood term $p(y_{1:T}|\mathcal{H}_{1:T})$ in (5.1) requires us to specify a generative process that gives rise to the observed audio samples. The prior term $p(\mathcal{H}_{1:T})$ reflects our knowledge about piano-rolls and other hidden variables. Our modelling task is therefore to specify both how, knowing the hidden variable states (essentially the piano-roll), the microphone samples will be generated, and also to state a prior on likely piano-rolls. Initially, we concentrate on the sound generation process of a single note.

5.2.1 Modelling a single note

Musical instruments tend to create oscillations with modes that are roughly related by integer ratios, albeit with strong damping effects and transient attack characteristics (Fletcher & Rossing, 1998). It is common to model such signals as the sum of a periodic component and a transient non-periodic component (See e.g. (Serra & Smith, 1991; Rodet, 1998; Irizarry, 2002)). The sinusoidal model (McAulay & Quatieri, 1986) is often a good approximation that provides a compact representation for the periodic component. The transient component can be modelled as a correlated Gaussian noise process (Parra & Jain, 2001; Davy & Godsill, 2003). Our signal model is also in the same spirit, but we will define it in state space form, because this provides a natural way to couple the signal model with the piano-roll representation. Similar formulations are used in the econometrics literature to model seasonal fluctuations, e.g. see (Harvey, 1989; West & Harrison, 1997). Here we omit the transient component and focus on the periodic component. It is conceptually straightforward to include the transient component as this does not effect the complexity of our inference algorithms.

First we consider how to generate a damped sinusoid y_t through time, with angular frequency ω . Consider a Gaussian process where typical realizations $y_{1:T}$ are damped “noisy” sinusoidal signals with angular frequency ω :

$$s_t \sim \mathcal{N}(\rho_t B(\omega)s_{t-1}, Q) \quad (5.2)$$

$$y_t \sim \mathcal{N}(Cs_t, R) \quad (5.3)$$

$$s_0 \sim \mathcal{N}(0, S) \quad (5.4)$$

$$B(\omega) = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \quad (5.5)$$

We use $\mathcal{N}(\mu, \Sigma)$ to denote a multivariate Gaussian distribution with mean μ and covariance Σ . Here $B(\omega)$ is a Givens rotation matrix that rotates two dimensional vector s_t by ω degrees counterclockwise. C is a projection matrix defined as $C = [1, 0]$. The phase and amplitude characteristics of y_t are determined by the initial condition s_0 drawn from a prior with covariance S . The damping factor $0 \leq \rho_t \leq 1$ specifies the rate at which s_t contracts to 0. See Figure 5.1 for an example. The transition noise variance Q is used to model deviations from an entirely deterministic linear model. The observation noise variance R models background noise.

In reality, musical instruments (with a definite pitch) have several modes of oscillation that are



Figure 5.1: A damped oscillator in state space form. Left: At each time step, the state vector s rotates by ω and its length becomes shorter. Right: The actual waveform is a one dimensional projection from the two dimensional state vector. The stochastic model assumes that there are two independent additive noise components that corrupt the state vector s and the sample y , so the resulting waveform $y_{1:T}$ is a damped sinusoid with both phase and amplitude noise.

roughly located at integer multiples of the fundamental frequency ω . We can model such signals by a bank of oscillators giving a block diagonal transition matrix $A_t = A(\omega, \rho_t)$ defined as

$$\begin{pmatrix} \rho_t^{(1)} B(\omega) & 0 & \dots & 0 \\ 0 & \rho_t^{(2)} B(2\omega) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \rho_t^{(H)} B(H\omega) \end{pmatrix} \quad (5.6)$$

where H denotes the number of *harmonics*, assumed to be known. To reduce the number of free parameters we define each harmonic damping factor $\rho^{(h)}$ in terms of a basic ρ . A possible choice is to take $\rho_t^{(h)} = \rho_t^h$, motivated by the fact that damping factors of harmonics in a vibrating string scale approximately geometrically with respect to that of the fundamental frequency, i.e. higher harmonics decay faster (Valimaki, Huopaniemi, Karjalainen, & Janosy, 1996). $A(\omega, \rho_t)$ is the transition matrix at time t and encodes the physical properties of the sound generator as a first order Markov Process. The rotation angle ω can be made time dependent for modelling pitch drifts or vibrato. However, in this paper we will restrict ourselves to sound generators that produce sounds with (almost) constant frequency. The state of the sound generator is represented by s_t , a $2H$ dimensional vector that is obtained by concatenation of all the oscillator states in (5.2).

5.2.2 From Piano-Roll to Microphone

A piano-roll is a collection of indicator variables $r_{j,t}$, where $j = 1 \dots M$ runs over sound generators (i.e. notes or “keys” of a piano) and $t = 1 \dots T$ runs over time. Each sound generator has a unique fundamental frequency ω_j associated with it. For example, we can choose ω_j such that we cover all notes of the tempered chromatic scale in a certain frequency range. This choice is arbitrary and for a finer pitch analysis a denser grid with smaller intervals between adjacent notes can be used.

Each indicator is binary, with values “sound” or “mute”. The essential idea is that, if previously muted, $r_{j,t-1} = \text{“mute”}$ an onset for the sound generator j occurs if $r_{j,t} = \text{“sound”}$. The generator continues to sound (with a characteristic damping decay) until it is again set to “mute”, when the generated signal decays to zero amplitude (much) faster. The piano-roll, being a collection of indicators $r_{1:M,1:T}$, can be viewed as a binary sequence, e.g. see Figure 5.2. Each row of the piano-roll $r_{j,1:T}$ controls an underlying sound generator.

The piano-roll determines the both sound onset generation, and the damping of the note. We consider first the damping effects.

Piano-Roll : Damping

Thanks to our simple geometrically related damping factors for each harmonic, we can characterise the damping factor for each note $j = 1, \dots, M$ by two decay coefficients ρ_{sound} and ρ_{mute} such that

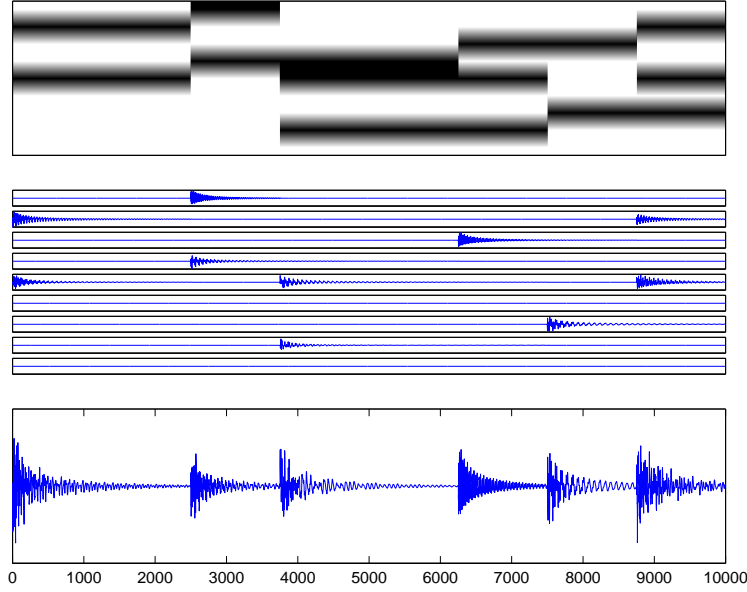


Figure 5.2: Piano-roll. The vertical axis corresponds to the sound generator index j and the horizontal axis corresponds to time index t . Black and white pixels correspond to “sound” and “mute” respectively. The piano-roll can be viewed as a binary sequence that controls an underlying signal process. Each row of the piano-roll $r_{j,1:T}$ controls a sound generator. Each generator is a Gaussian process (a Kalman filter model), where typical realizations are damped periodic waveforms of a constant fundamental frequency. As in a piano, the fundamental frequency is a function of the generator index j . The actual observed signal $y_{1:T}$ is a superposition of the outputs of all generators.

$1 \geq \rho_{\text{sound}} > \rho_{\text{mute}} > 0$. The piano-roll $r_{j,1:T}$ controls the damping coefficient $\rho_{j,t}$ of note j at time t by:

$$\rho_{j,t} = \rho_{\text{sound}}[r_{j,t} = \text{sound}] + \rho_{\text{mute}}[r_{j,t} = \text{mute}] \quad (5.7)$$

Here, and elsewhere in the article, the notation $[x = \text{text}]$ has value equal to 1 when variable x is in state text, and is zero otherwise. We denote the transition matrix as $A_j^{\text{mute}} \equiv A(\omega_j, \rho_{\text{mute}})$; similarly for A_j^{sound} .

Piano-Roll : Onsets

At each new onset, i.e. when $(r_{j,t-1} = \text{mute}) \rightarrow (r_{j,t} = \text{sound})$, the old state s_{t-1} is “forgotten” and a new state vector is drawn from a Gaussian prior distribution $\mathcal{N}(0, S)$. This models the energy injected into a sound generator at an onset (this happens, for example, when a guitar string is plucked). The amount of energy injected is proportional to the determinant of S and the covariance structure of S describes how this total energy is distributed among the harmonics. The covariance matrix S thus captures some of the timbre characteristics of the sound. The transition and observation equations are given by

$$\text{isonset}_{j,t} = (r_{j,t-1} = \text{mute} \wedge r_{j,t} = \text{sound}) \quad (5.8)$$

$$A_{j,t} = [r_{j,t} = \text{mute}]A_j^{\text{mute}} + [r_{j,t} = \text{sound}]A_j^{\text{sound}} \quad (5.9)$$

$$s_{j,t} \sim \begin{aligned} &[-\text{isonset}_{j,t}]\mathcal{N}(A_{j,t}s_{t-1}, Q) \\ &+ [\text{isonset}_{j,t}]\mathcal{N}(0, S) \end{aligned} \quad (5.10)$$

$$y_{j,t} \sim \mathcal{N}(Cs_{j,t}, R) \quad (5.11)$$

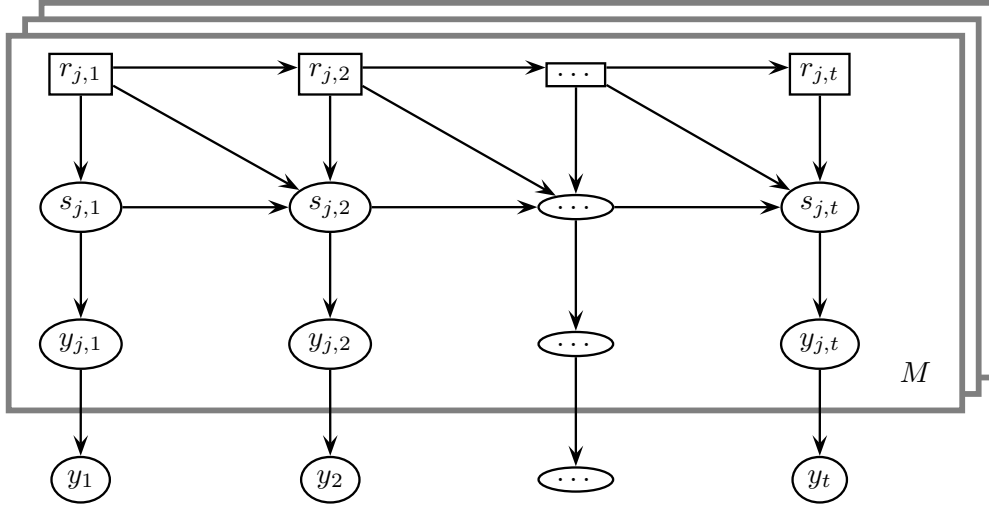


Figure 5.3: Graphical Model. The rectangle box denotes “plates”, M replications of the nodes inside. Each plate, $j = 1, \dots, M$ represents the sound generator (note) variables through time.

In the above, C is a $1 \times 2H$ projection matrix $C = [1, 0, 1, 0, \dots, 1, 0]$ with zero entries on the even components. Hence $y_{j,t}$ has a mean being the sum of the damped harmonic oscillators. R models the variance of the noise in the output of each sound generator. Finally, the observed audio signal is the superposition of the outputs of all sound generators,

$$y_t = \sum_j y_{j,t} \quad (5.12)$$

The generative model (5.7)-(5.12) can be described qualitatively by the graphical model in Figure 5.3. Equations (5.11) and (5.12) define $p(y_{1:T} | s_{1:M,1:T})$. Equations (5.7) (5.9) and (5.10) relate r and s and define $p(s_{1:M,1:T} | r_{1:M,1:T})$. In this paper, the prior model $p(r_{1:M,1:T})$ is Markovian and has the following factorial structure¹:

$$p(r_{1:M,1:T}) = \prod_m \prod_t p(r_{m,t} | r_{m,t-1})$$

5.2.3 Inference

Given the polyphonic model described in section 5.2, to infer the most likely piano-roll we need to compute

$$r_{1:M,1:T}^* = \operatorname{argmax}_{r_{1:M,1:T}} p(r_{1:M,1:T} | y_{1:T}) \quad (5.13)$$

where the posterior is given by

$$p(r_{1:M,1:T} | y_{1:T}) = \frac{1}{p(y_{1:T})} \int_{s_{1:M,1:T}} p(y_{1:T} | s_{1:M,1:T}) \\ \times p(s_{1:M,1:T} | r_{1:M,1:T}) p(r_{1:M,1:T})$$

¹In the simulations we have fixed the transition parameter $p(r = \text{mute} | r = \text{sound}) = p(r = \text{sound} | r = \text{mute}) = 10^{-7}$

The normalization constant, $p(y_{1:T})$, obtained by summing the integral term over all configurations $r_{1:M,1:T}$ is called the evidence.²

Unfortunately, calculating this most likely piano-roll configuration is generally intractable, and is related to the difficulty of inference in Switching Kalman Filters (Murphy, 1998, 2002). We shall need to develop approximation schemes for this general case, to which we shall return in a later section.

As a prelude, we consider a slightly simpler, related model which aims to track the pitch (melody identification) in a monophonic instrument (playing only a single note at a time), such as a flute. The insight gained here in the inference task will guide us to a practical approximate algorithm in the more general case later.

5.3 Monophonic Model

Melody identification, or monophonic pitch tracking with onset and offset detection, can be formulated by a small modification of our general framework. Even this simplified task is still of huge practical interest, e.g. in real time MIDI conversion for controlling digital synthesizers using acoustical instruments or pitch tracking from the singing voice. One important problem in real time pitch tracking is the time/frequency tradeoff: to estimate the frequency accurately, an algorithm needs to collect statistics from a sufficiently long interval. However, this often conflicts with the real time requirements.

In our formulation, each sound generator is a dynamical system with a sequence of transition models, sound and mute. The state s evolves first according to the sounding regime with transition matrix A^{sound} and then according to the muted regime with A^{mute} . The important difference from a general switching Kalman filter is that when the indicator r switches from mute to sound, the old state vector is “forgotten”. By exploiting this fact, in the appendix 5.6.1 we derive, for a single sound generator (i.e. a single note of a fixed pitch that gets on and off), an exact polynomial time algorithm for calculating the evidence $p(y_{1:T})$ and MAP configuration $r_{1:T}^*$.

Monophonic pitch tracking

Here we assume that at any given time t only a single sound generator can be sounding, i.e. $r_{j,t} = \text{sound} \Rightarrow r_{j',t} = \text{mute}$ for $j' \neq j$. Hence, for practical purposes, the factorial structure of our original model is redundant; i.e. we can “share” a single state vector s among all sound generators³. The resulting model will have the same graphical structure as a single sound generator but with an indicator $j_t \in 1 \dots M$ which indexes the active sound generator, and $r_t \in \{\text{sound}, \text{mute}\}$ indicates sound or mute. Inference for this case turns out to be also tractable (i.e. polynomial). We allow

²It is instructive to interpret (5.13) from a Bayesian model selection perspective (MacKay, 2003). In this interpretation, we view the set of all piano-rolls, indexed by configurations of discrete indicator variables $r_{1:M,1:T}$, as the set of all models among which we search for the best model $r_{1:M,1:T}^*$. In this view, state vectors $s_{1:M,1:T}$ are the model parameters that are integrated over. It is well known that the conditional predictive density $p(y|r)$, obtained through integration over s , automatically penalizes more complex models, when evaluated at $y = y_{1:T}$. In the context of piano-roll inference, this objective will automatically prefer solutions with less notes. Intuitively, this is simply because at each note onset, the state vector s_t is reinitialized using a broad Gaussian $\mathcal{N}(0, S)$. Consequently, a configuration r with more onsets will give rise to a conditional predictive distribution $p(y|r)$ with a larger covariance. Hence, a piano-roll that claims the existence of additional onsets without support from data will get a lower likelihood.

³We ignore the cases when two or more generators are simultaneously in the mute state.

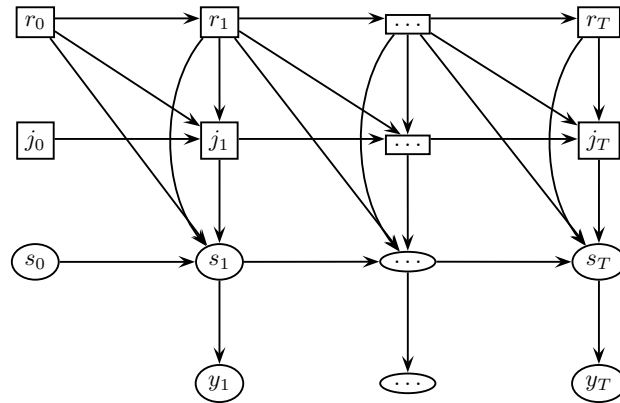


Figure 5.4: Simplified Model for monophonic transcription. Since there is only a single sound generator active at any given time, we can represent a piano-roll at each time slice by the tuple (j_t, r_t) where j_t is the index of the active sound generator and $r_t \in \{\text{sound}, \text{mute}\}$ indicates the state.

switching to a new j' only after an onset. The full generative model using the pairs (j_t, r_t) , which includes both likelihood and prior terms is given as

$$\begin{aligned}
 r_t &\sim p(r_t | r_{t-1}) \\
 \text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\
 j_t &\sim [\neg \text{isonset}_t] \delta(j_t; j_{t-1}) + [\text{isonset}_t] u(j_t) \\
 A_t &= [r_t = \text{mute}] A_{j_t}^{\text{mute}} + [r_t = \text{sound}] A_{j_t}^{\text{sound}} \\
 s_t &\sim [\neg \text{isonset}_t] \mathcal{N}(A_t s_{t-1}, Q) + [\text{isonset}_t] \mathcal{N}(0, S) \\
 y_t &\sim \mathcal{N}(C s_t, R)
 \end{aligned}$$

Here $u(j)$ denotes a uniform distribution on $1, \dots, M$ and $\delta(j_t; j_{t-1})$ denotes a degenerate (deterministic) distribution concentrated on j_t , i.e. unless there is an onset the active sound generator stays the same. Our choice of a uniform $u(j)$ simply reflects the fact that any new note is as likely as any other. Clearly, more informative priors, e.g. that reflect knowledge about tonality, can also be proposed. Similarly, for doing a more precise pitch analysis, we may choose a finer grid such that $\omega_{j+1}/\omega_j = \mathcal{Q}$. Here, \mathcal{Q} is the quality factor, a measure of the desired frequency precision not to be confused with the transition noise Q .

The graphical model is shown in Figure 5.4. The derivation of the polynomial time inference algorithm is given in appendix 5.6.2. Technically, it is a simple extension of the single note algorithm derived in appendix 5.6.1.

In Figure 5.5, we illustrate the results on synthetic data sampled from the model where we show the filtering density $p(r_t, j_t | y_{1:t})$. After an onset, the posterior becomes quickly crisp, long before we observe a complete cycle. This feature is especially attractive for real time applications where a reliable pitch estimate has to be obtained as early as possible.

We conclude this subsection with an illustration on real data. We have recorded a major scale on an electric bass and downsampled from the original sampling rate of $F_s = 22050$ by a factor of $D = 10$. We have estimated parameters for a signal model with $H = 8$ harmonics. The “training set” consisted of a single note recorded from the same instrument; this procedure will be discussed in more detail in section 5.5. We have estimated the MAP configuration $(r, j)_{1:T}$ using the algorithm described in appendix 5.6.2. The figure shows that the estimated piano roll is quite precise. We have repeated the experiment on a pianoroll with a pitch grid of $1/4$ semitones

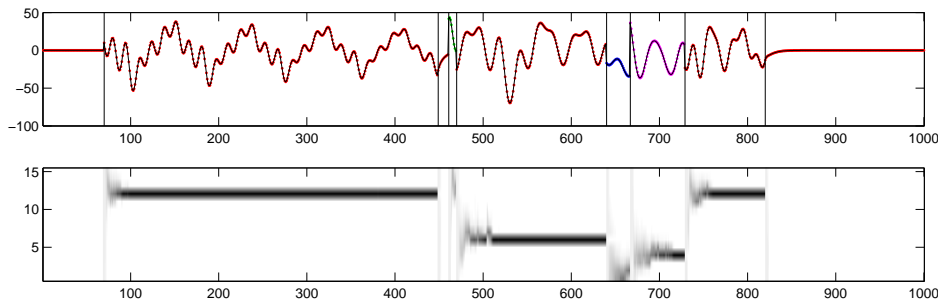


Figure 5.5: Monophonic pitch tracking. (Top) Synthetic data sampled from model in Figure 5.4. Vertical bars denote the onset and offset times. (Bottom) The filtering density $p(r_t, j_t | y_{1:t})$. The vertical axis denotes the sound generator index j_t and the gray level denotes the posterior probability $p(r_t = \text{sound}, j_t | y_{1:t})$ where black corresponds to 1.

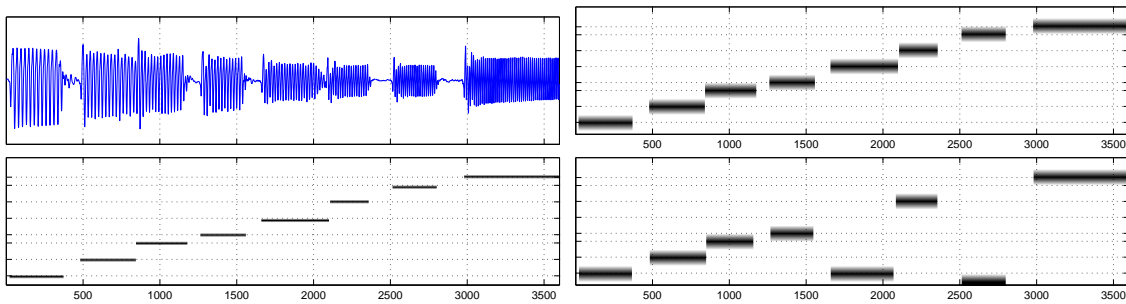


Figure 5.6: Monophonic pitch estimation on real data. (Top, left) F major scale played on an electric bass. (Top, right) Estimated MAP configuration $(r, j)_{1:T}$. (Bottom, left) A finer analysis with $Q = 2^{1/48}$ reveals that the 5'th and 7'th degree of the scale are intonated slightly low. (Bottom, right) Poorer results may be obtained when signal model parameters are not set correctly.

($Q = 2^{1/48}$). The results reveal that the 5'th and 7'th degree of the scale were intonated slightly low, which didn't had much effect on the estimation of the pitch class when using a coarser grid. In the last experiment we have trained the model parameters using a note sung by a vocalist. As expected, the results are poorer; in particular we observe that 5'ths or octaves are confused due to the different harmonic structure and transition characteristics.

Extension to vibrato and legato

The monophonic model has been constructed such that the rotation angle ω remains constant. Although the the transition noise with variance Q still allows for small and independent deviations in frequencies of the harmonics, the model is not realistic for situations with systematic pitch drift or fluctuation, e.g. as is the case with vibrato. Moreover, on many musical instruments, it is possible to play *legato*, that is without an explicit onset between note boundaries. In our

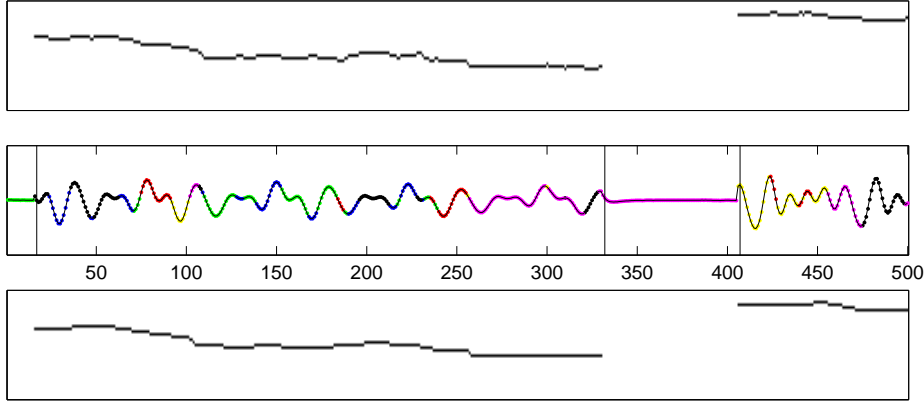


Figure 5.7: Tracking varying pitch. Top and middle panel show the true piano-roll and the sampled signal. The estimated piano-roll is shown below.

framework, pitch drift and legato can be modelled as a sequence of transition models. Consider the generative process for the note index j :

$$\begin{aligned}
 r_t &\sim p(r_t|r_{t-1}) \\
 \text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\
 \text{issound}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{sound}) \\
 j_t &\sim [\text{issound}_t]d(j_t|j_{t-1}) + \\
 &\quad [r_t = \text{mute}]\delta(j_t; j_{t-1}) + [\text{isonset}_t]u(j_t)
 \end{aligned}$$

Here, $d(j_t|j_{t-1})$ is a multinomial distribution reflecting our prior belief how likely is it to switch between notes. When $r_t = \text{mute}$, there is no regime change, reflected by the deterministic distribution $\delta(j_t; j_{t-1})$ peaked around j_{t-1} . Remember that neighbouring notes have also close fundamental frequency ω . To simulate pitch drift, we choose a fine grid such that $\omega_j/\omega_{j+1} = \mathcal{Q}$. In this case, we can simply define $d(j_t|j_{t-1})$ as a multinomial distribution with support on $[j_{t-1} - 1, j_{t-1}, j_{t-1} + 1]$ with cell probabilities $[d_{-1} \ d_0 \ d_1]$. We can take a larger support for $d(j_t|j_{t-1})$, but in practice we would rather reduce the frequency precision \mathcal{Q} to avoid additional computational cost.

Unfortunately, the terms included by the drift mechanism render an exact inference procedure intractable. We derive the details of the resulting algorithm in the appendix 5.6.2. A simple deterministic pruning method is described in appendix 5.6.2. In Figure 5.7, we show the estimated MAP trajectory $r_{1:T}^*$ for drifting pitch. We use a model where the quality factor is $\mathcal{Q} = 2^{-120}$, (120 generators per octave) with drift probability $d_{-1} = d_1 = 0.1$. A fine pitch contour, that is accurate to sample precision, can be estimated.

5.4 Polyphonic Inference

In this section we return to the central goal of inference in the general polyphonic model described in section 5.2. To infer the most likely piano-roll we need to compute $\underset{r_{1:M,1:T}}{\operatorname{argmax}} p(r_{1:M,1:T}|y_{1:T})$ defined in (5.13). Unfortunately, the calculation of (5.13) is intractable. Indeed, even the calculation of the Gaussian integral conditioned on a particular configuration $r_{1:M,1:T}$ using standard Kalman filtering equations is prohibitive since the dimension of the state vector is $|s| = 2H \times M$, where H is the number of harmonics. For a realistic application we may have $M \approx 50$ and $H \approx 10$. It is

clear that unless we are able to develop efficient approximation techniques, the model will be only of theoretical interest.

5.4.1 Vertical Problem: Chord identification

Chord identification is the simplest polyphonic transcription task. Here we assume that a given audio signal $y_{1:T}$ is generated by a piano-roll where $r_{j,t} = r_j$ for all⁴ $j = 1 \dots M$. The task is to find the MAP configuration

$$r_{1:M}^* = \operatorname{argmax}_{r_{1:M}} p(y_{1:T}, r_{1:M})$$

Each configuration corresponds to a chord. The two extreme cases are “silence” and “cacophony” that correspond to configurations $r_{1:M}[\text{mute} \text{ mute} \dots \text{mute}]$ and $[\text{sound} \text{ sound} \dots \text{sound}]$ respectively. The size of the search space in this case 2^M , which is prohibitive for direct computation.

A simple approximation is based on greedy search: we start iterative improvement from an initial configuration $r_{1:M}^{(0)}$ (silence, or randomly drawn from the prior). At each iteration i , we evaluate the probability $p(y_{1:T}, r_{1:M})$ of all neighbouring configurations of $r_{1:M}^{(i-1)}$. We denote this set by $\text{neigh}(r_{1:M}^{(i-1)})$. A configuration $r' \in \text{neigh}(r)$, if r' can be reached from r within a single flip (i.e., we add or remove single notes). If $r_{1:M}^{(i-1)}$ has a higher probability than all its neighbours, the algorithm terminates, having found a local maximum. Otherwise, we pick the neighbour with the highest probability and set

$$r_{1:M}^{(i)} = \operatorname{argmax}_{r_{1:M} \in \text{neigh}(r_{1:M}^{(i-1)})} p(y_{1:T}, r_{1:M})$$

and iterate until convergence. We illustrate the algorithm on a signal sampled from the generative model, see Figure 5.8. This procedure is guaranteed to converge to a (possibly local) maxima. Nevertheless, we observe that for many examples this procedure is able to identify the correct chord. Using multiple restarts from different initial configurations will improve the quality of the solution at the expense of computational cost.

One of the advantages of our generative model based approach is that we can in principle infer a chord given any subset of data. For example, we can simply downsample $y_{1:T}$ (without any preprocessing) by an integer factor of D and view the discarded samples as missing values. Of course, when D is large, i.e. when we throw away many samples, due to diminishing likelihood contribution, we obtain a diffuse posterior on the piano-roll and eventually the results will be poorer.

In Figure 5.9, we show the results of such an experiment. We have downsampled $y_{1:T}$ with factor $D = 2, 3$ and 4 . The energy spectrum is quite coarse due to the short length of the data. Consequently many harmonics are not resolved, e.g. we can not identify the underlying line spectrum by visual inspection. Methods based on template matching or identification of peaks may have serious problems for such examples. On the other hand, our model driven approach is able to identify the true chord. We note that, the presented results are illustrative only and the actual behaviour of the algorithm (sensitivity to D , importance of starting configuration) will depend on the details of the signal model.

⁴We will assume that initially we start from silence where $r_{j,0} = \text{mute}$ for all $j = 1 \dots M$

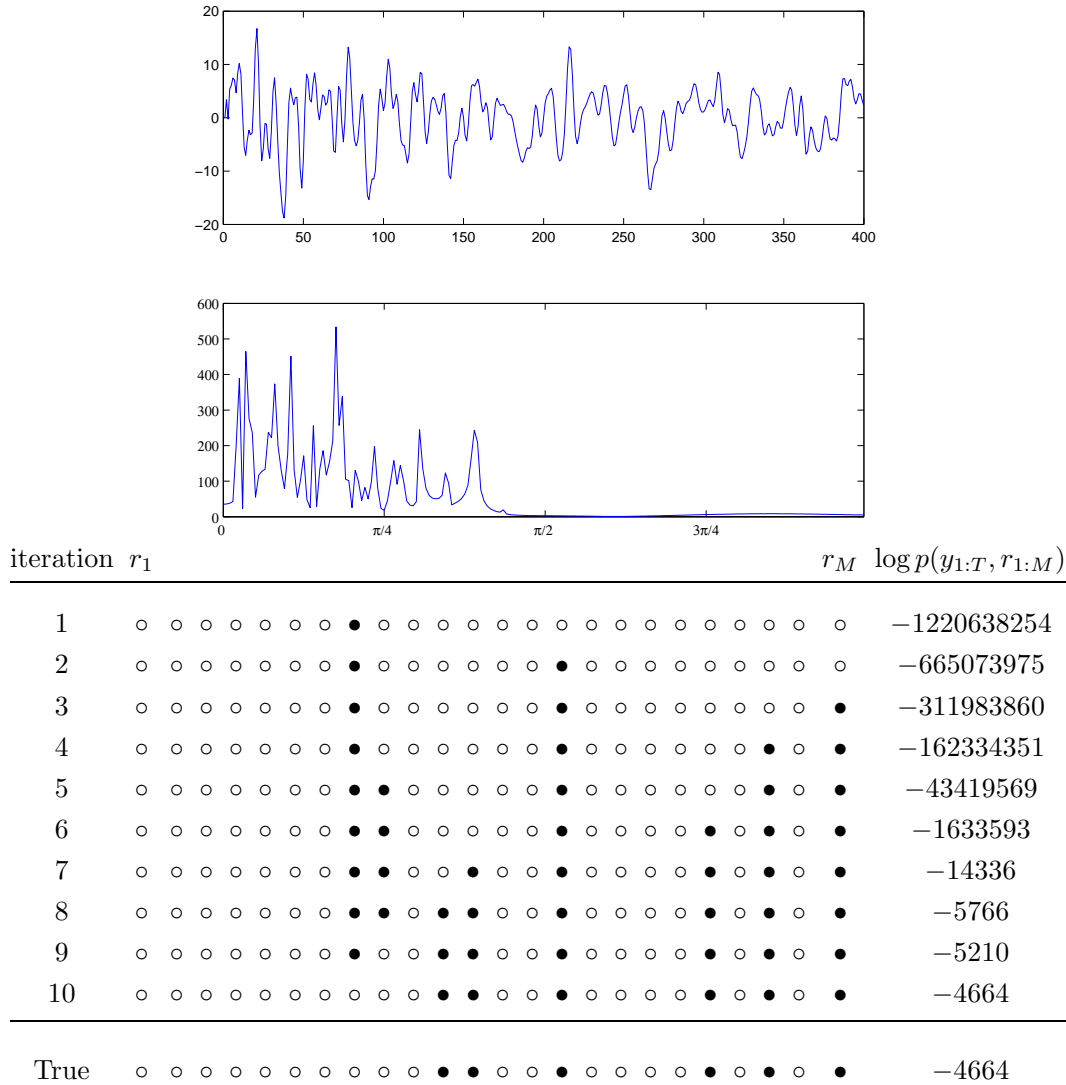
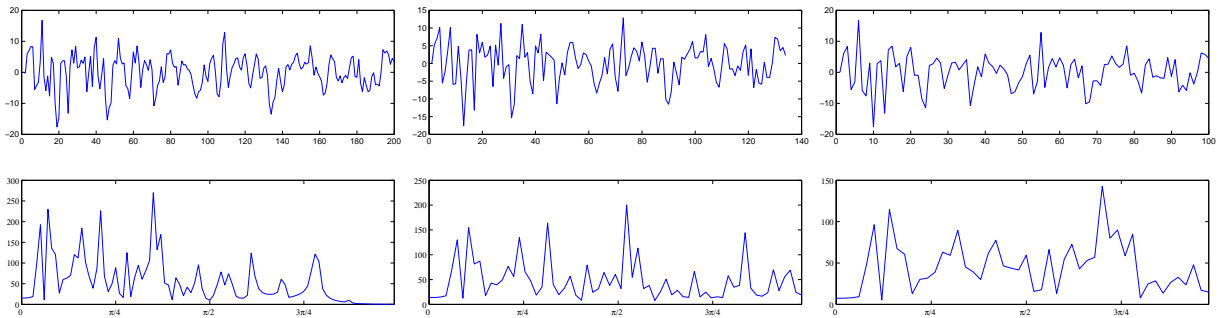


Figure 5.8: We have first drawn a random piano-roll configuration (a random chord) $r_{1:M}$. Given $r_{1:M}$, we generate a signal of length 400 samples with a sampling frequency $F_s = 4000$ from $p(y_{1:T}|r_{1:M})$. We assume 24 notes (2 octaves). The synthesized signal from the generative model and its discrete time Fourier transform modulus are shown above. The true chord configuration and the associated log probability is at the bottom of the table. For the iterative algorithm, the initial configuration in this example was silence. At this point we compute the probability for each single note configurations (all one flip neighbours of silence). The first note that is added is actually not present in the chord. Until iteration 9, all iterations add extra notes. Iteration 9 and 10 turn out to be removing the extra notes and iterations converge to the true chord. The intermediate configurations visited by the algorithm are shown in the table below. Here, sound and mute states are represented by ●'s and ○'s.



D		$p(y_{1:D:T}, r_{1:M})$	Init
2	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ●	-2685	True
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ● ● ○ ○ ● ○ ● ○ ● ○ ●	-3179	Silence
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ●	-2685	Random
3	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ●	-2057	True
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ●	-2057	Silence
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ● ○ ● ○ ○ ● ● ● ● ○ ○ ●	-2616	Random
4	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ●	-1605	True
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ● ○ ○ ● ○ ● ○ ○ ● ○ ○ ○	-1668	Silence
	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ● ○ ○ ● ○ ○ ○ ● ○ ● ○ ○	-1591	Random

Figure 5.9: Iterative improvement results when data are subsampled by a factor of $D = 2, 3$ and 4 , respectively. For each factor D , the top line shows the true configuration and the corresponding probability. The second line is the solution found by starting from silence and the third line is starting from a random configuration drawn from the prior (best of 3 independent runs).

5.4.2 Piano-Roll inference Problem: Joint Chord and Melody identification

The piano-roll estimation problem can be viewed as an extension of chord identification in that we also detect onsets and offsets for each note within the analysis frame. A practical approach is to analyze the signal in sufficiently short time windows and assume that for each note, at most one changepoint can occur within the window.

Consider data in a short window, say $y_{1:W}$. We start iterative improvement from a configuration $r_{1:M,1:W}^{(0)}$, where each time slice $r_{1:M,t}^{(0)}$ for $t = 1 \dots W$ is equal to a ‘‘chord’’ $r_{1:M,0}$. The chord $r_{1:M,0}$ can be silence or, during a frame by frame analysis, the last time slice of the best configuration found in the previous analysis window. Let the configuration at $i - 1$ ’th iteration be denoted as $r_{1:M,1:W}^{(i-1)}$. At each new iteration i , we evaluate the posterior probability $p(y_{1:W}, r_{1:M,1:W})$, where $r_{1:M,1:W}$ runs over all neighbouring configuration of $r_{1:M,1:W}^{(i-1)}$. Each member $r_{1:M,1:W}$ of the neighbourhood is generated as follows: For each $j = 1 \dots M$, we clamp all the other rows, i.e. we set $r_{j',1:W} = r_{j',1:W}^{(i-1)}$ for $j' \neq j$. For each time step $t = 1 \dots W$, we generate a new configuration such that the switches up to time t are equal to the initial switch $r_{j,0}$, and its opposite $\neg r_{j,0}$ after t , i.e. $r_{j,t} = r_{j,0}[t' < t] + \neg r_{j,0}[t' \geq t]$. This is equivalent to saying that a sounding note may get muted, or a muted note may start to sound. The computational advantage of allowing only one changepoint at each row is that the probability of all neighbouring configurations for a fixed j can be computed by a single backward, forward pass (Cemgil & Kappen, 2003; Murphy, 2002). Finally, we pick the neighbour with the maximum probability. The algorithm is illustrated in Figure 5.10.

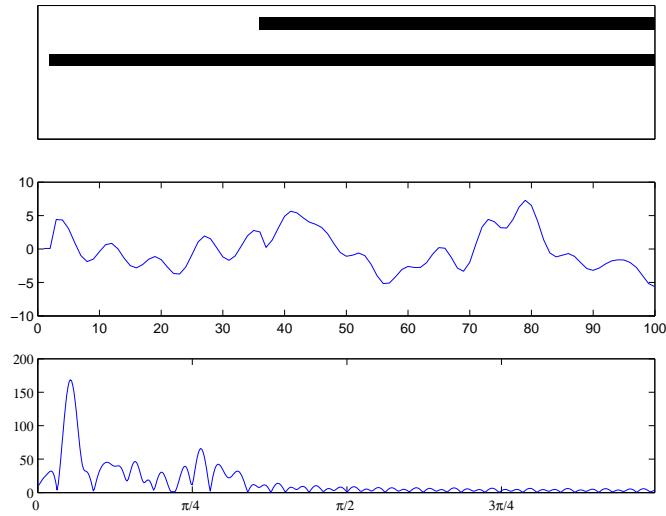
The analysis for the whole sequence proceeds as follows: Consider two successive analysis windows $Y_{\text{prev}} \equiv y_{1:W}$ and $Y \equiv y_{W+1:2W}$. Suppose we have obtained a solution $R_{\text{prev}}^* \equiv r_{1:M,1:W}^*$ obtained by iterative improvement. Conditioned on R_{prev}^* , we compute the posterior $p(s_{1:M,W} | Y_{\text{prev}}, R_{\text{prev}}^*)$ by Kalman filtering. This density is the prior of s for the current analysis window Y . The search starts from a chord equal to the last time slice of R_{prev}^* . In Fig. 5.11 we show an illustrative result obtained by this algorithm on synthetic data. In similar experiments with synthetic data, we are often able to identify the correct piano-roll.

This simple greedy search procedure is somewhat sensitive to location of onsets within the analysis window. Especially, when an onset occurs near the end of an analysis window, it may be associated with an incorrect pitch. The correct pitch is often identified in the next analysis window, when a longer portion of the signal is observed. However, since the basic algorithm does not allow for correcting the previous estimate by retrospection, this introduces some artifacts. A possible method to overcome this problem is to use a fixed lag smoothing approach, where we simply carry out the analysis on overlapping windows. For example, for an analysis window $Y_{\text{prev}} \equiv y_{1:W}$, we find $r_{1:M,1:W}^*$. The next analysis window is taken as $y_{L+1:W+L}$ where $L \leq W$. We find the prior $p(s_{1:M,L} | y_{1:L}, r_{1:M,1:L}^*)$ by Kalman filtering. On the other hand, obviously, the algorithm becomes slower by a factor of L/W .

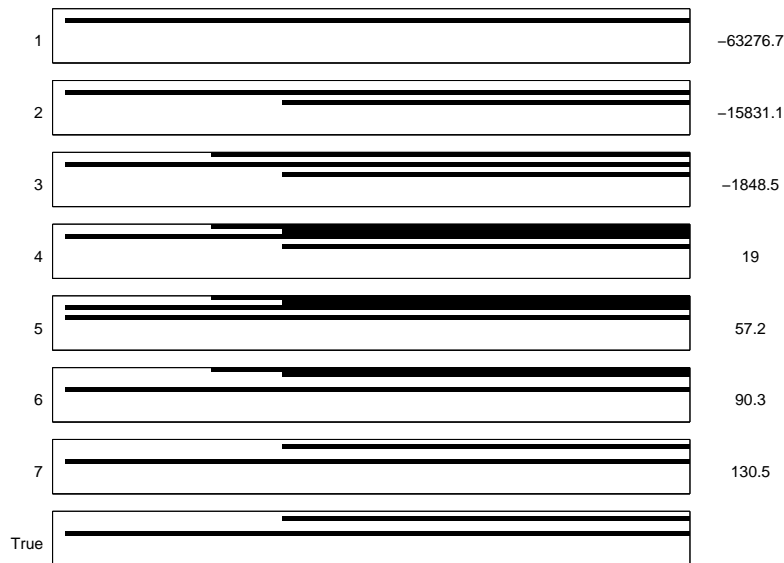
An optimal choice for L and W will depend upon many factors such as signal characteristics, sampling frequency, downsampling factor D , onset/offset positions, number of active sound generators at a given time as well as the amount of CPU time available. In practice, these values may be critical and they need to be determined by trial and error. On the other hand, it is important to note that L and W just determine how the approximation is made but not enter the underlying model.

5.5 Learning

In the previous sections, we assumed that the correct signal model parameters $\theta = (S, \rho, Q, R)$ were known. These include in particular the damping coefficients $\rho_{\text{sound}}, \rho_{\text{mute}}$, transition noise variance Q , observation noise R and the initial prior covariance matrix S after an onset. In practice,



(a)



(b)

Figure 5.10: Iterative improvement with changepoint detection. The true piano-roll, the signal and its Fourier transform magnitude are shown in Figure 5.10.(a). In Figure 5.10.(b), configurations $r^{(i)}$ visited during iterative improvement steps. Iteration numbers i are shown left and the corresponding probability is shown on the right. The initial configuration (i.e. “chord”) $r_{1:M,0}$ is set to silence. At the first step, the algorithm searches all single note configurations with a single onset. The winning configuration is shown on top panel of Figure 5.10.(b). At the next iteration, we clamp the configuration for this note and search in a subset of two note configurations. This procedure adds and removes notes from the piano-roll and converges to a local maxima. Typically, the convergence is quite fast and the procedure is able to identify the true chord without making a “detour” as in (b).

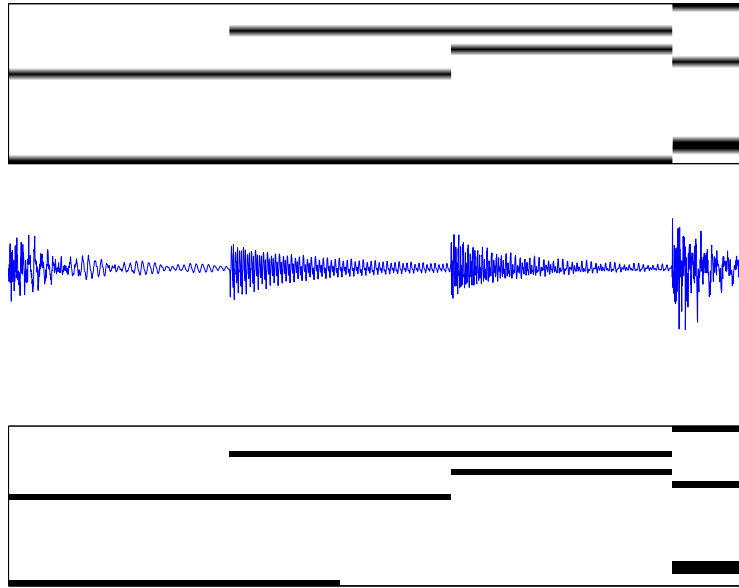


Figure 5.11: A typical example for Polyphonic piano-roll inference from synthetic data. We generate a realistic piano-roll (top) and render a signal using the polyphonic model (middle). Given only the signal, we estimate the piano-roll by iterative improvement in successive windows (bottom). In this example, only the offset time of the lowest note is not estimated correctly. This is a consequence that, for long notes, the state vector s converges to zero before the generator switches to the mute state.

for an instrument class (e.g. plucked string instruments) a reasonable range for θ can be specified a-priori. We may safely assume that θ will be static (not time dependent) during a given performance. However, exact values for these quantities will vary among different instruments (e.g. old and new strings) and recording/performance conditions.

One of the well-known advantages of Bayesian inference is that, when uncertainty about parameters is incorporated in a model, this leads in a natural way to the formulation of a learning algorithm. The piano-roll estimation problem, omitting the time indices, can be stated as follows:

$$r^* = \operatorname{argmax}_r \int_{\theta} \int_s p(y|s, \theta) p(s|r, \theta) p(\theta) p(r) \quad (5.14)$$

In other words, we wish to find the best piano-roll by taking into account all possible settings of the parameter θ , weighted by the prior. Note that (5.14) becomes equivalent to (5.13), if we knew the “best” parameter θ^* , i.e. $p(\theta) = \delta(\theta - \theta^*)$. Unfortunately, the integration on θ can not be calculated analytically and approximation methods must be used (Ghahramani & Beal, 2000). A crude but computationally cheap approximation replaces the integration on θ in (5.14) with maximization:

$$r^* = \operatorname{argmax}_r \max_{\theta} \int_s p(y|s, \theta) p(s|r, \theta) p(\theta) p(r)$$

Essentially, this is a joint optimization problem on piano-rolls and parameters which we solve by a greedy coordinate ascent algorithm. The algorithm we propose is a double loop algorithm where

we iterate in the outer loop between maximization over r and maximization over θ . The latter maximization itself is calculated with an iterative algorithm EM.

$$\begin{aligned} r^{(i)} &= \operatorname{argmax}_r \int_s p(y|s, \theta^{(i-1)})p(s|r, \theta^{(i-1)})p(\theta^{(i-1)})p(r) \\ \theta^{(i)} &= \operatorname{argmax}_\theta \int_s p(y|s, \theta)p(s|r^{(i)}, \theta)p(\theta)p(r^{(i)}) \end{aligned}$$

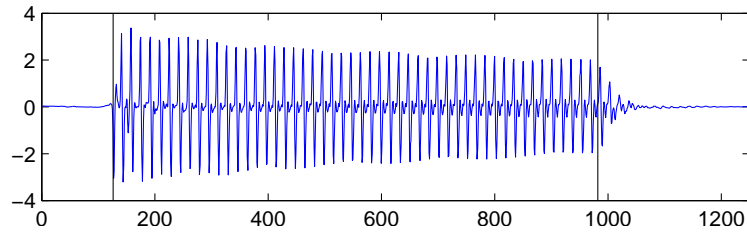
For a single note, conditioned on a fixed $\theta^{(i-1)}$, $r^{(i)}$ can be calculated exactly, using the message propagation algorithm derived in appendix 5.6.2. Conditioned on $r^{(i)}$, maximization on the θ coordinate becomes equivalent to parameter estimation in linear dynamical systems, for which no closed form solution is known. Nevertheless, this step can be calculated by an iterative expectation maximization (EM) algorithm (Murphy, 2002; Ghahramani & Hinton, 1996). In practice, we observe that for realistic starting conditions $\theta^{(0)}$, the $r^{(i)}$ are identical, suggesting that the best segmentation r^* is not very sensitive to variations in θ near to a local optimum. In Figure 5.12, we show the results of training the signal model based on a single note (a C from the low register) of an electric bass.

In an experiment with real data, we illustrate the performance of the model for two and three note polyphony (See Fig.5.13). We have recorded three separate monophonic melodies; ascending modes of the major scale starting from the root, 3'rd and 5'th degree of a major scale. We have estimated model parameters using a single note from the same register. For each monophonic melody, we have calculated the ground truth $r_{1:M,1:T}^{\text{true}}$ by the algorithm described in section 5.3. We have constructed the two note example by adding the first two melodies. The analysis is carried out using a window length of $W = 200$ samples, without overlap between analysis frames (i.e. $L = W$). We were able to identify the correct pitch classes for the two note polyphony case. However, especially some note offsets are not detected correctly. In the three note case, pitch classes are correct, but there are also more artifacts, e.g. the chord around sample index 500 is identified incorrect. We expect results to go worse with increasing polyphony; this behaviour is qualitatively similar to other methods reported in the literature, e.g. (Sterian, 1999; Walmsley, 2000), but clearly, more simulation studies have to be carried out for an objective comparison. Investigating the loglikelihood ratio $\log \frac{p(y_{1:T}|r_{1:M,1:T}^{\text{true}})p(r_{1:M,1:T}^{\text{true}})}{p(y_{1:T}|r_{1:M,1:T}^*)p(r_{1:M,1:T}^*)} \gg 0$ suggests that the failure is due to the suboptimal estimation procedure, i.e. the model prefers the true solution but our greedy algorithm is unable to locate it and gets stuck in $r_{1:M,1:T}^*$, where r^* denotes here the configuration found by the algorithm. In the conclusions section, we will discuss some alternative approximation methods to improve results.

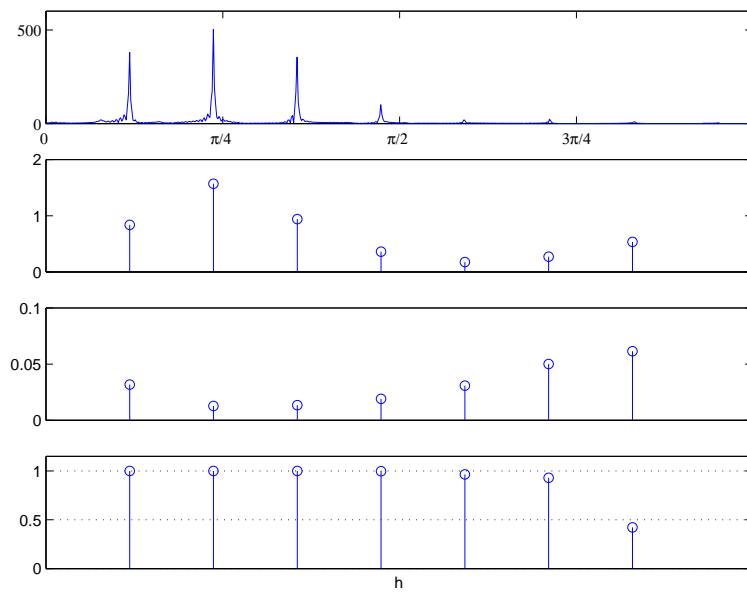
5.6 Discussion

We have presented a model driven approach where transcription is viewed as a Bayesian inference problem. In this respect, at least, our approach parallels the previous work of Walmsley (2000), Davy and Godsill (2003), Raphael (2002). We believe, however, that our formulation, based on a switching state space model, has several advantages. We can remove the assumption of a frame based model and this enables us to analyse music online and to sample precision. Practical approximations to an eventually intractable exact posterior can be carried out frame-by-frame, such as by using a fixed time-lag smoother. This, however, is merely a computational issue (albeit a very important one). We may also discard samples to reduce computational burden, and account for this correctly in our model.

An additional advantage of our formulation is that we can still deliver a pitch estimate even when the fundamental and lower harmonics of the frequency band are missing. This is related to

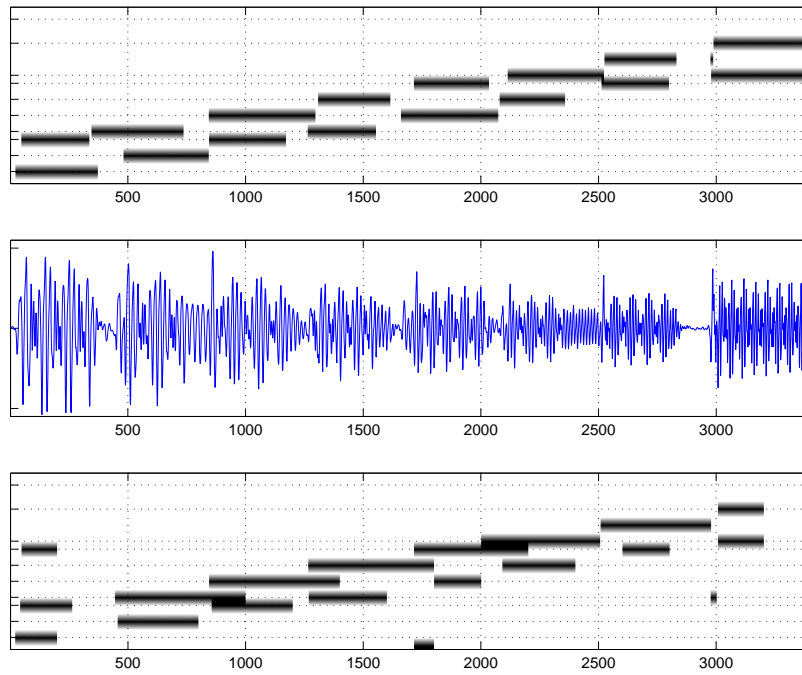


(a) A single note from an electric bass. Original sampling rate of 22050 Hz is reduced by downsampling with factor $D = 20$. Vertical lines show the changepoints of the MAP trajectory $r_{1:K}$.

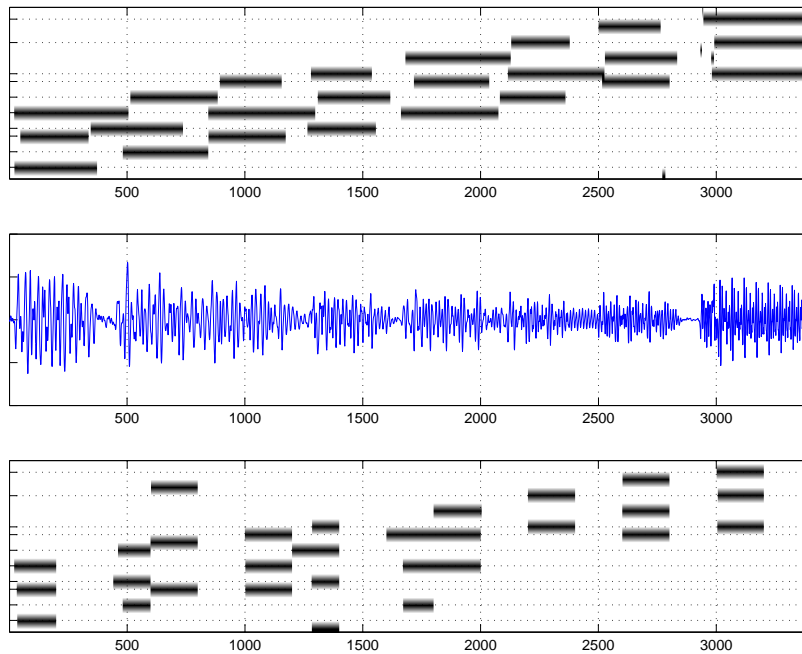


(b) Top to Bottom: Fourier transform of the downsampled signal and diagonal entries of S , Q and damping coefficients ρ_{sound} for each harmonic.

Figure 5.12: Training the signal model with EM from a single note from an electric bass using a sampling rate of 22050 Hz. The original signal is downsampled by a factor of $D = 20$. Given some crude first estimate for model parameters $\theta^{(0)}(S, \rho, Q, R)$, we estimate $r^{(1)}$, shown in (a). Conditioned on $r^{(1)}$, we estimate the model parameters $\theta^{(1)}$ and so on. Let S_h denote the 2×2 block matrix from the diagonal S , corresponding to the h 'th harmonic, similarly for Q_h . In (b), we show the estimated parameters for each harmonic sum of diagonal elements, i.e. $\text{Tr } S_h$ and $\text{Tr } Q_h$. The damping coefficient is found as $\rho_{\text{sound}} = (\det A_h A_h^T)^{1/4}$ where A_h is a 2×2 diagonal block matrix of transition matrix A^{sound} . For reference, we also show the Fourier transform modulus of the downsampled signal. We can see, that on the low frequency bands, S mimics the average energy distribution of the note. However, transient phenomena, such as the strongly damped 7'th harmonic with relatively high transition noise, is hardly visible in the frequency spectrum. On the other hand for online pitch detection, such high frequency components are important to generate a crisp estimate as early as possible.



(a) (Top) The ground truth estimated when all melodies are transcribed separately. (Middle) The superposition of melodies downsampled by a factor of $D = 10$. (Bottom) Piano-roll estimated with an analysis window of size $W = 200$ samples, without overlap between analysis frames



(b) Result for the same experiment with three notes polyphony.

Figure 5.13: Experiment with two and three note polyphony.

so called *virtual pitch* perception (Terhardt, 1974): we tend to associate notes with a pitch class depending on the relationship between harmonics rather than the frequency of the fundamental component itself.

There is a strong link between model selection and polyphonic music transcription. In chord identification we need to compare models with different number of notes, and in melody identification we need to deduce the number of onsets. Model selection becomes conceptually harder when one needs to compare models of different size. We partially circumvent this difficulty by using switch variables, which implicitly represent the number of components.

Following the established signal processing jargon, we may call our approach a time-domain method, since we are not explicitly calculating a discrete-time Fourier transform. On the other hand, the signal model presented here has close links to the Fourier analysis and sinusoidal modelling. Our analysis can be interpreted as a search procedure for a sparse representation on a set of basis vectors. In contrast to Fourier analysis, where the basis vectors are sinusoids (e.g. see (Qi, Minka, & Picard, 2002) for a Bayesian treatment), we represent the observed signal implicitly using signals drawn from a stochastic process which typically generates decaying periodic oscillations (e.g. notes) with occasional changepoints. The sparsity of this representation is a consequence of the onset mechanism, that effectively puts a mixture prior over the hidden state vector s . This prior is peaked around zero and has broad tails, indicating that most of the sources are muted and only a few are sounding. It is well known that such Gaussian mixture priors induce sparse representations, e.g. see (Attias, 1999; Olshausen & Millman, 2000) for applications in the context of source separation.

5.6.1 Future work

Although our approach has many desirable features (automatically deducing number of correct notes, high temporal resolution e.t.c.), one of the main disadvantage of our method is computational cost associated with updating large covariance matrices in Kalman filtering. It would be very desirable to investigate approximation schemas that employ fast transformations such as the FFT to accelerate computations.

When transcribing music, human experts rely heavily on prior knowledge about the musical structure – harmony, tempo or expression. Such structure can be captured by training probabilistic generative models on a corpus of compositions and performances by collecting statistics over selected features (e.g. (Raphael & Stoddard, 2003)). One of the important advantages of our approach is that such prior knowledge about the musical structure can be formulated as an informative prior on a piano-roll; thus can be integrated in signal analysis in a consistent manner. We believe that investigation of this direction is important in designing robust and practical music transcription systems.

Our signal model considered here is inspired by additive synthesis. An advantage of our linear formulation is that we can use the Kalman filter recursions to integrate out the continuous latent state analytically. An alternative would be to formulate a nonlinear dynamical system that implements a nonlinear synthesis model (e.g. FM synthesis, waveshaping synthesis, or even a physical model (Smith, 1992)). Such an approach would reduce the dimensionality of the latent state space but force us to use approximate integration methods such as particle filters or EKF/UKF (Doucet et al., 2001). It remains an interesting open question whether, in practice, one should trade-off analytical tractability versus reduced latent state dimension.

In this paper, for polyphonic transcription, we have used a relatively simple deterministic inference method based on iterative improvement. The basic greedy algorithm, whilst still potentially useful in practice, may get stuck in poor solutions. We believe that, using our model as a framework, better polyphonic transcriptions can be achieved using more elaborate inference or search methods. For example, computation time associated with exhaustive search of the neighbourhood

for all visited configurations could be significantly reduced by randomizing the local search (e.g. by Metropolis-Hastings moves) or use heuristic proposal distributions derived from easy-to-compute features such as the energy spectrum. Alternatively, sequential Monte Carlo methods or deterministic message propagation algorithms such as Expectation propagation (EP) (Minka, 2001) could be also used.

We have not yet tested our model for more general scenarios, such as music fragments containing percussive instruments or bell sounds with inharmonic spectra. Our simple periodic signal model would be clearly inadequate for such a scenario. On the other hand, we stress the fact that the framework presented here is not only limited to the analysis of signals with harmonic spectra, and in principle applicable to any family of signals that can be represented by a switching state space model. This is already a large class since many real-world acoustic processes can be approximated well with piecewise linear regimes. We can also formulate a joint estimation schema for unknown parameters as in (5.14) and integrate them out (e.g. see Davy and Godsill (2003)). However, this is currently a hard and computationally expensive task. If efficient and accurate approximate integration methods can be developed, our model will be applicable to mixtures of many different types of acoustical signals and may be useful in more general auditory scene analysis problems.

Appendix 5.A Derivation of message propagation algorithms

In the appendix, we derive several exact message propagation algorithms. Our derivation closely follows the standard derivation of recursive prediction and update equations for the Kalman filter (Bar-Shalom & Li, 1993). First we focus on a single sound generator. In appendix 5.6.1 and 5.6.2, we derive polynomial time algorithms for calculating the evidence $p(y_{1:T})$ and MAP configuration $r_{1:T}^* = \operatorname{argmax}_{r_{1:T}} p(y_{1:T}, r_{1:T})$ respectively. The MAP configuration is useful for onset/offset detection. In the following section, we extend the onset/offset detection algorithms to monophonic pitch tracking with constant frequency. We derive a polynomial time algorithm for this case in appendix 5.6.2. The case for varying fundamental frequency is derived in the following appendix 5.6.2. In appendix 5.6.2 we describe heuristics to reduce the amount of computations.

5.A.1 Computation of the evidence $p(y_{1:T})$ for a single sound generator by forward filtering

We assume a Markovian prior on the indicators r_t where $p(r_t = i | r_{t-1} = j) \equiv p_{i,j}$. For convenience, we repeat the generative model for a single sound generator by omitting the note index j .

$$\begin{aligned} r_t &\sim p(r_t | r_{t-1}) \\ \text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\ s_t &\sim [\neg \text{isonset}_t] \mathcal{N}(A_{r_t} s_{t-1}, Q) + [\text{isonset}_t] \mathcal{N}(0, S) \\ y_t &\sim \mathcal{N}(C s_t, R) \end{aligned}$$

For simplicity, we will sometime use the labels 1 and 2 to denote sound and mute respectively. We enumerate the transition models as $f_{r_t}(s_t | s_{t-1}) = \mathcal{N}(A_{r_t} s_{t-1}, Q)$. We define the filtering potential as

$$\alpha_t \equiv p(y_{1:t}, s_t, r_t, r_{t-1}) = \sum_{r_{1:t-2}} \int_{s_{0:t-1}} p(y_{1:t}, s_{0:t}, r_{1:t})$$

We assume that y is always observed, hence we use the term potential to indicate the fact that $p(y_{1:t}, s_t, r_t, r_{t-1})$ is not normalized. The filtering potential is in general a conditional Gaussian mixture, i.e. a mixture of Gaussians for each configuration of $r_{t-1:t}$. We will highlight this data structure by using the following notation

$$\alpha_t \equiv \left\{ \begin{array}{cc} \alpha_t^{1,1} & \alpha_t^{1,2} \\ \alpha_t^{2,1} & \alpha_t^{2,2} \end{array} \right\}$$

where each $\alpha_t^{i,j} = p(y_{1:t}, s_t, r_t = i, r_{t-1} = j)$ for $i, j = 1 \dots 2$ are also Gaussian mixture potentials. We will denote the conditional normalization constants as

$$Z_t^i \equiv p(y_{1:t}, r_t = i) = \sum_{r_{t-1}} \int_{s_t} \alpha_t^{i,r_{t-1}}$$

Consequently the evidence is given by

$$Z_t \equiv p(y_{1:t}) = \sum_{r_t} \sum_{r_{t-1}} \int_{s_t} \alpha_t = \sum_i Z_t^i$$

We also define the predictive density

$$\begin{aligned} \alpha_{t|t-1} &\equiv p(y_{1:t-1}, s_t, r_t, r_{t-1}) \\ &= \sum_{r_{t-2}} \int_{s_{t-1}} p(s_t | s_{t-1}, r_t, r_{t-1}) p(r_t | r_{t-1}) \alpha_{t-1} \end{aligned}$$

In general, for switching Kalman filters, calculating exact posterior features, such as the evidence $Z_t = p(y_{1:t})$, is not tractable. This is a consequence of the fact that the number of mixture components required to represent the exact filtering density α_t grows exponentially with time step k (i.e. one Gaussian for each of the exponentially many configurations $r_{1:t}$). Luckily, for the model we are considering here, the growth is polynomial in k only. See also (Fearnhead, 2003).

To see this, suppose we have the filtering density available at time $t-1$ as α_{t-1} . The transition models can be organized also in a table where i 'th row and j 'th column correspond to $p(s_t | s_{t-1}, r_t = i, r_{t-1} = j)$

$$p(s_t | s_{t-1}, r_t, r_{t-1}) = \left\{ \begin{array}{cc} f_1(s_t | s_{t-1}) & \pi(s_t) \\ f_2(s_t | s_{t-1}) & f_2(s_t | s_{t-1}) \end{array} \right\}$$

Calculation of the predictive potential is straightforward. First, summation over r_{t-2} yields

$$\sum_{r_{t-2}} \alpha_{t-1} = \left\{ \begin{array}{cc} \alpha_{t-1}^{1,1} + \alpha_{t-1}^{1,2} \\ \alpha_{t-1}^{2,1} + \alpha_{t-1}^{2,2} \end{array} \right\} \equiv \left\{ \begin{array}{c} \xi_{t-1}^1 \\ \xi_{t-1}^2 \end{array} \right\}$$

Integration over s_{t-1} and multiplication by $p(r_t | r_{t-1})$ yields the predictive potential

$$\alpha_{t|t-1} = \left\{ \begin{array}{cc} p_{1,1} \psi_1^1(s_t) & p_{1,2} Z_{t-1}^2 \pi(s_t) \\ p_{2,1} \psi_2^1(s_t) & p_{2,2} \psi_2^2(s_t) \end{array} \right\}$$

where we define

$$Z_{t-1}^2 \equiv \int_{s_{t-1}} \xi_{t-1}^2 \quad \psi_i^j(s_t) \equiv \int_{s_{t-1}} f_i(s_t | s_{t-1}) \xi_{t-1}^j$$

The potentials ψ_i^j can be computed by applying the standard Kalman prediction equations to each component of ξ_{t-1}^j . The updated potential is given by $\alpha_t = p(y_t|s_t)\alpha_{t|t-1}$. This quantity can be computed by applying standard Kalman update equations to each component of $\alpha_{t|t-1}$.

From the above derivation, it is clear that $\alpha_t^{1,2}$ has only a single Gaussian component. This has the consequence that the number of Gaussian components in $\alpha_t^{1,1}$ increases only linearly (the first row-sum terms ξ_{t-1}^1 propagated through f_1). The second row sum term ξ_t^2 is more costly; it increases at every time slice by the number of components in ξ_{t-1}^1 . Since the size of ξ_{t-1}^1 grows linearly, the size of ξ_t^2 grows quadratically with time t .

5.6.2 Computation of MAP configuration $r_{1:T}^*$

The MAP state is defined as

$$\begin{aligned} r_{1:T}^* &= \operatorname{argmax}_{r_{1:T}} \int_{s_{0:T}} p(y_{1:T}, s_{0:T}, r_{1:T}) \\ &\equiv \operatorname{argmax}_{r_{1:T}} \int_{s_{0:T}} \phi(s_{0:T}, r_{1:T}) \end{aligned}$$

For finding the MAP state, we replace summations over r_t by maximization. One potential technical difficulty is that, unlike in the case for evidence calculation, maximization and integration do not commute. Consider a conditional Gaussian potential

$$\phi(s, r) \equiv \{\phi(s, r = 1), \phi(s, r = 2)\}$$

where $\phi(s, r)$ are Gaussian potentials for each configuration of r . We can compute the MAP configuration

$$r^* = \operatorname{argmax}_r \int_s \phi(s, r) = \operatorname{argmax} \{Z^1, Z^2\}$$

where $Z^j = \int_s \phi(s, r = j)$. We evaluate the normalization of each component (i.e. integrate over the continuous hidden variable s first) and finally find the maximum of all normalization constants.

However, direct calculation of $r_{1:T}^*$ is not feasible because of exponential explosion in the number of distinct configurations. Fortunately, for our model, we can introduce a deterministic pruning schema that reduces the number of kernels to a polynomial order and meanwhile guarantees that we will never eliminate the MAP configuration. This exact pruning method hinges on the factorization of the posterior for the assignment of variables $r_t = 1, r_{t-1} = 2$ (mute to sound transition) that breaks the direct link between s_t and s_{t-1} :

$$\begin{aligned} \phi(s_{0:T}, r_{1:t-2}, r_{t-1} = 2, r_t = 1, r_{t+1:T}) &= \\ \phi(s_{0:t-1}, r_{1:t-2}, r_{t-1} = 2) \phi(s_{t:T}, r_{t+1:T}, r_t = 1 | r_{t-1} = 2) \end{aligned} \quad (5.15)$$

In this case:

$$\begin{aligned} &\max_{r_{1:T}} \int_{s_{0:T}} \phi(s_{0:T}, r_{1:t-2}, r_{t-1} = 2, r_t = 1, r_{t+1:T}) \\ &= \max_{r_{1:t-1}} \int_{s_{0:t-1}} \phi(s_{0:t-1}, r_{1:t-2}, r_{t-1} = 2) \\ &\quad \times \max_{r_{t:T}} \int_{s_{t:T}} \phi(s_{t:T}, r_{t+1:T}, r_t = 1 | r_{t-1} = 2) \\ &= Z_t^2 \times \max_{r_{t+1:T}} \int_{s_{t:T}} \phi(s_{t:T}, r_{t+1:T}, r_t = 1 | r_{t-1} = 2) \end{aligned} \quad (5.16)$$

This Equation shows that whenever we have an onset, we can calculate the maximum over the past and future configurations separately. Put differently, provided that the MAP configuration has the form $r_{1:T}^* = [r_{1:t-3}^*, r_{t-1} = 2, r_t = 1, r_{t+1:T}^*]$, the prefix $[r_{1:t-3}^*, r_{t-1} = 2]$ will be the solution for the reduced maximization problem $\operatorname{argmax}_{r_{1:t-1}} \int_{s_{0:t-1}} \phi(s_{0:t-1}, r_{1:t-1})$.

Forward pass

Suppose we have a collection of Gaussian potentials

$$\delta_{t-1} \equiv \left\{ \begin{array}{cc} \delta_{t-1}^{1,1} & \delta_{t-1}^{1,2} \\ \delta_{t-1}^{2,1} & \delta_{t-1}^{2,2} \end{array} \right\} \equiv \left\{ \begin{array}{c} \delta_{t-1}^1 \\ \delta_{t-1}^2 \end{array} \right\}$$

with the property that the Gaussian kernel corresponding the prefix $r_{1:t-1}^*$ of the MAP state is a member of δ_{t-1} , i.e. $\phi(s_{t-1}, r_{1:t-1}^*) \in \delta_{t-1}$ s.t. $r_{1:T}^* = [r_{1:t-1}^*, r_{t:T}^*]$. We also define the subsets

$$\begin{aligned} \delta_{t-1}^{i,j} &= \{\phi(s_{t-1}, r_{1:t-1}) : \phi \in \delta_{t-1} \text{ and } r_{t-1} = i, r_{t-2} = j\} \\ \delta_{t-1}^i &= \bigcup_j \delta_{t-1}^{i,j} \end{aligned}$$

We show how we find δ_t . The prediction is given by

$$\delta_{t|t-1} = \int_{s_{t-1}} p(s_t | s_{t-1}, r_t, r_{t-1}) p(r_t | r_{t-1}) \delta_{t-1}$$

The multiplication by $p(r_t | r_{t-1})$ and integration over s_{t-1} yields the predictive potential $\delta_{t|t-1}$

$$\left\{ \begin{array}{cc} p_{1,1} \int_{s_{t-1}} f_1(s_t | s_{t-1}) \delta_{t-1}^1 & p_{1,2} \pi(s_t) \int_{s_{t-1}} \delta_{t-1}^2 \\ p_{2,1} \int_{s_{t-1}} f_2(s_t | s_{t-1}) \delta_{t-1}^1 & p_{2,2} \int_{s_{t-1}} f_2(s_t | s_{t-1}) \delta_{t-1}^2 \end{array} \right\}$$

By the (5.16), we can replace the collection of numbers $\int_{s_{t-1}} \delta_{t-1}^2$ with with the scalar $Z_{t-1}^2 \equiv \max \int_{s_{t-1}} \delta_{t-1}^2$ without changing the optimum solution:

$$\delta_{t|t-1}^{1,2} = p_{1,2} Z_{t-1}^2 \pi(s_t)$$

The updated potential is given by $\delta_t = p(y_t | s_t) \delta_{t|t-1}$. The analysis of the number of kernels proceeds as in the previous section.

Decoding

During the forward pass, we tag each Gaussian component of δ_t with its past history of $r_{1:t}$. The MAP state can be found by a simple search in the collection of polynomially many numbers and reporting the associated tag:

$$r_{1:T}^* = \operatorname{argmax}_{r_{1:T}} \int_{s_T} \delta_T$$

We finally conclude that the forward filtering and MAP (Viterbi path) estimation algorithms are essentially identical with summation replaced by maximization and an additional tagging required for decoding.

5.A.3 Inference for monophonic pitch tracking

In this section we derive an exact message propagation algorithm for monophonic pitch tracking. Perhaps surprisingly, inference in this case turns out to be still tractable. Even though the size of the configuration space $r_{1:M,1:T}$ is of size $(M+1)^T = O(2^{T \log M})$, the space complexity of an exact algorithm remains quadratic in t . First, we define a ‘‘mega’’ indicator node $z_t = (j_t, r_t)$ where

$j_t \in 1 \dots M$ indicates the index of the active sound generator and $r_t \in \{\text{sound, mute}\}$ indicates its state. The transition model $p(z_t|z_{t-1})$ is a large sparse transition table with probabilities

$$\left(\begin{array}{ccc|ccc} p_{1,1} & & & p_{1,2}/M & \dots & p_{1,2}/M \\ & \ddots & & \vdots & \ddots & \vdots \\ & & p_{1,1} & p_{1,2}/M & \dots & p_{1,2}/M \\ \hline p_{2,1} & & & p_{2,2} & & \\ & \ddots & & & \ddots & \\ & & p_{2,1} & & & p_{2,2} \end{array} \right) \quad (5.17)$$

where the transitions $p(z_t = (j, r)|z_{t-1} = (j', r'))$ are organized at the n 'th row and m 'th column where $n = r \times M + j - 1$ and $m = r' \times M + j' - 1$. (5.17). The transition models $p(s_t|s_{t-1}, z_t = (j, r), z_{t-1} = (j', r'))$ can be organized similarly:

$$\left(\begin{array}{ccc|ccc} f_{1,1} & & & \pi(s_t) & \dots & \pi(s_t) \\ & \ddots & & \vdots & \ddots & \vdots \\ & & f_{1,M} & \pi(s_t) & \dots & \pi(s_t) \\ \hline f_{2,1} & & & f_{2,1} & & \\ & \ddots & & & \ddots & \\ & & f_{2,M} & & & f_{2,M} \end{array} \right)$$

Here, $f_{r,j} \equiv f_{r,j}(s_t|s_{t-1})$ denotes the transition model of the j 'th sound generator when in state r . The derivation for filtering follows the same lines as the onset/offset detection model, with only slightly more tedious indexing. Suppose we have the filtering density available at time $t - 1$ as α_{t-1} . We first calculate the predictive potential. Summation over z_{t-2} yields the row sums

$$\xi_{t-1}^{(r,j)} = \sum_{r',j'} \alpha_{t-1}^{(r,j),(r',j')}$$

Integration over s_{t-1} and multiplication by $p(z_t|z_{t-1})$ yields the predictive potential $\alpha_{t|t-1}$. The components are given as

$$\alpha_{t|t-1}^{(r,j)(r',j')} = \begin{cases} (1/M)p_{r,r'}\pi(s_t)Z_{t-1}^{(r',j')} & r = 1 \wedge r' = 2 \\ [j = j'] \times p_{r,r'}\psi_t^{(r,j)(r',j')} & \text{otherwise} \end{cases} \quad (5.18)$$

where we define

$$Z_{t-1}^{(r',j')} \equiv \int_{s_{t-1}} \xi_{t-1}^{(r',j')} \\ \psi_t^{(r,j)(r',j')} \equiv \int_{s_{t-1}} f_{r,j}(s_t|s_{t-1})\xi_{t-1}^{(r',j')}$$

The potentials ψ can be computed by applying the standard Kalman prediction equations to each component of ξ . Note that the forward messages have the same sparsity structure as the prior, i.e. $\alpha_{t-1}^{(r,j)(r',j')} \neq 0$ when $p(r_t = r, j_t = j|r_{t-1} = r', j_t = j')$ is nonzero. The updated potential is given by $\alpha_t = p(y_t|s_t)\alpha_{t|t-1}$. This quantity can be computed by applying standard Kalman update equations to each nonzero component of $\alpha_{t|t-1}$.

5.A.4 Monophonic pitch tracking with varying fundamental frequency

We model pitch drift by a sequence of transition models. We choose a grid such that $\omega_j/\omega_{j+1} = \mathcal{Q}$, where \mathcal{Q} is close to one. Unfortunately, the subdiagonal terms introduced to the prior transition matrix $p(z_t = (1, j_t) | z_{t-1} = (1, j_{t-1}))$

$$p_{1,1} \times \begin{pmatrix} (d_0 + d_1) & d_{-1} & & & \\ d_1 & d_0 & d_{-1} & & \\ & d_1 & \ddots & \ddots & \\ & & \ddots & d_0 & d_{-1} \\ & & & d_1 & (d_0 + d_{-1}) \end{pmatrix} \quad (5.19)$$

render an exact algorithm exponential in t . The recursive update equations, starting with α_{t-1} , are obtained by summing over z_{t-2} , integration over s_{t-1} and multiplication by $p(z_t | z_{t-1})$. The only difference is that the prediction equation (5.18) needs to be changed to

$$\alpha_{t|t-1}^{(r,j)(r',j')} = \begin{cases} d(j-j') \times p_{r,r'} \psi_t^{(r,j)(r',j')} & r = 1 \wedge r' = 1 \\ (1/M) p_{r,r'} \pi(s_t) Z_{t-1}^{(r',j')} & r = 1 \wedge r' = 2 \\ [j = j'] \times p_{r,r'} \psi_t^{(r,j)(r',j')} & r = 2 \end{cases}$$

where ψ and Z are defined in (5.19). The reason for the exponential growth is the following: Remember that each $\psi^{(r,j)(r',j')}$ has as many components as an entire row sum of $\xi_{t-1}^{(r,j)} = \sum_{r',j'} \alpha_{t-1}^{(r,j)(r',j')}$. Unlike the inference for piecewise constant pitch estimation, now at some rows there are two or more messages (e.g. $\alpha_{t|t-1}^{(1,j)(1,j)}$ and $\alpha_{t|t-1}^{(1,j)(1,j+1)}$) that depend on ψ .

Appendix 5.B Computational Simplifications

5.B.1 Pruning

Exponential growth in message size renders an algorithm useless in practice. Even in special cases, where the message size increases only polynomially in T , this growth is still prohibitive for many applications. A cheaper approximate algorithm can be obtained by pruning the messages. To keep the size of messages bounded, we limit the number of components to N and store only components with the highest evidence. An alternative is discarding components of a message that contribute less than a given fraction (e.g. 0.0001) to the total evidence. More sophisticated pruning methods with profound theoretical justification, such as resampling (Cemgil & Kappen, 2003) or collapse (Heskes, 2002), are viable alternatives but these are computationally more expensive. In our simulations, we observe that using a simple pruning method with the maximum number of components per message set to $N = 100$, we can obtain results very close to an exact algorithm.

5.B.2 Kalman filtering in a reduced dimension

Kalman filtering with a large state dimension $|s|$ at typical audio sampling rates $F_s \approx 40$ kHz may be prohibitive with generic hardware. This problem becomes more severe when the number of notes M is large, (which is typically around 50 – 60), than even conditioned on a particular configuration $r_{1:M}$, the calculation of the filtering density is expensive. Hence, in an implementation, tricks of precomputing the covariance matrices can be considered (Bar-Shalom & Li, 1993) to further reduce the computational burden.

Another important simplification is less obvious from the graphical structure and is a consequence of the inherent asymmetry between the sound and mute states. Typically, when a note switches and stays for a short period in the mute state, i.e. $r_{j,t} = \text{mute}$ for some period, the marginal posterior over the state vector $s_{j,t}$ will converge quickly to a zero mean Gaussian with a small covariance matrix *regardless* of observations y . We exploit this property to save computations by clamping the hidden states for sequences of $s_{j,t:t'}$ to zero for $r_{j,t:t'} = \text{“mute”}$. This reduces the hidden state dimension, since typically, only a few sound generators will be in sound state.

Publications

International refereed journals

- A. T. Cemgil, H. J. Kappen, and D. Barber. *A generative model for music transcription*. Accepted to IEEE Transactions on Speech and Audio Processing, 2004.
- A. T. Cemgil and H. J. 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, P. Desain, and H. Honing. *On tempo tracking: Tempogram representation and Kalman filtering*. Journal of New Music Research, 28:4:259-273, 2001.
- A. T. Cemgil, P. Desain, and H. J. Kappen. *Rhythm quantization for transcription*. Computer Music Journal, 24:2:60-76, 2000.

Selected Conference Proceedings

- A. T. Cemgil, H. J. Kappen, and D. Barber. Generative model based polyphonic music transcription. In Proc. of IEEE WASPAA, New Paltz, NY, October 2003. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.
- A.T. Cemgil, D. Barber, and H. J. Kappen. A dynamical bayesian network for tempo and polyphonic pitch tracking. In Proceedings of ICANN, Istanbul/Turkey, 2003.
- A.T. Cemgil and H. J. Kappen. Integrating tempo tracking and quantization using particle filtering. In Proceedings of the 2002 International Computer Music Conference, pages 419-422, Gothenburg/Sweden, 2002.
- A. T. Cemgil and H. J. Kappen. Rhythm quantization and tempo tracking by sequential Monte Carlo. In Thomas G. Dietterich, Sue Becker, and Zoubin Ghahramani, editors, Advances in Neural Information Processing Systems 14, pages 1361-1368. MIT Press, 2002.
- A.T. Cemgil and H. J. Kappen. Bayesian real-time adaptation for interactive performance systems. In Proceedings of the 2001 International Computer Music Conference, pages 147-150, Habana/Cuba, 2001.
- A.T. Cemgil, H. J. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and kalman filtering. In Proceedings of the 2000 International Computer Music Conference, pages 352-355, Berlin, 2000. (This paper has received the Swets and Zeitlinger Distinguished Paper Award of the ICMC 2000).

- A. T. Cemgil, P. Desain, and H. J. Kappen. Rhythm quantization for transcription. In Proceedings of the AISB'99 Symposium on Musical Creativity, pages 140-146, Edinburgh, UK, April 1999. AISB.
- A.T Cemgil and C. Erkut. Calibration of physical models using artificial neural networks with application to plucked string instruments. In Proceedings of ISMA97, International Symposium on Musical Acoustics, Edinburgh UK, 1997.
- A. T. Cemgil and E. Caglar, H. Anarim. Comparison of wavelet filters for pitch detection of monophonic music signals. In Proceedings of European Conference on Circuit Theory and Design, (ECCTD95), 1995.

Bibliography

- Aarts, E. H. L., & van Laarhoven, P. J. M. (1985). Statistical cooling: A general approach to combinatorial optimization problems. *Philips Journal of Research*, 40(4), 193–226.
- Agon, C., Assayag, G., Fineberg, J., & Rueda, C. (1994). Kant: A critique of pure quantification. In *Proceedings of the International Computer Music Conference*, pp. 52–9, Aarhus, Denmark. International Computer Music Association.
- Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2002). An introduction to MCMC for machine learning. *Machine Learning*, to appear.
- Andrieu, C., & Doucet, A. (1999). Joint bayesian model selection and estimation of noisy sinusoids via reversible jump mcmc. *IEEE Trans. on Signal Processing*, 47(10), 2667–2676.
- Attias, H. (1999). Independent factor analysis. *Neural Computation*, 11(4), 803–851.
- Bar-Shalom, Y., & Fortmann, T. E. (1988). *Tracking and Data Association*. Academic Press.
- Bar-Shalom, Y., & Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Boston.
- Bello, J. (2003). *Towards the Automated Analysis of simple polyphonic music: A knowledge-based approach*. Ph.D. thesis, King's College London - Queen Mary, University of London.
- Bregman, A. (1990). *Auditory Scene Analysis*. MIT Press.
- Brown, G. J., & Cooke, M. (1994). Computational auditory scene analysis. *Computer Speech and Language*, 8(2), 297–336.
- Cambouropoulos, E. (2000). From MIDI to traditional musical notation. In *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, Austin, Texas.
- Carter, C. K., & Kohn, R. (1996). Markov Chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83(3), 589–601.
- Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemas. *Biometrika*, 83, 81–94.
- Casey, M. A. (1998). *Auditory Group Theory with Applications to Statistical Basis Methods for Structured Audio*. Ph.D. thesis, MIT Media Laboratory, Cambridge MA.
- Cemgil, A. T., Desain, P., & Kappen, H. J. (2000). Rhythm quantization for transcription. *Computer Music Journal*, 24:2, 60–76.
- Cemgil, A. T., & Kappen, H. J. (2002). Rhythm quantization and tempo tracking by sequential Monte Carlo. In Dietterich, T. G., Becker, S., & Ghahramani, Z. (Eds.), *Advances in Neural Information Processing Systems 14*, pp. 1361–1368. MIT Press.
- Cemgil, A. T., & Kappen, H. J. (2003). Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18, 45–81.

- Cemgil, A. T., Kappen, H. J., & Barber, D. (2003). Generative model based polyphonic music transcription. In *Proc. of IEEE WASPAA*, New Paltz, NY. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.
- Cemgil, A. T., Kappen, H. J., Desain, P., & Honing, H. (2001). On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28:4, 259–273.
- Chen, R., & Liu, J. S. (2000). Mixture Kalman filters. *J. R. Statist. Soc.*, 10.
- Clarke, E. F. (1985). Structure and expression in rhythmic performance. In Howell, P., Cross, I., & West, R. (Eds.), *Musical structure and cognition*. Academic Press, Inc., London.
- Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons, Inc., New York.
- Dannenberg, R. (1984). An on-line algorithm for real-time accompaniment. In *Proceedings of ICMC*, pp. 193–198, San Francisco.
- Dannenberg, R. (1993). Music understanding by computer. In *Proceedings of the International Workshop on Knowledge Technology in the Arts*.
- Davy, M., & Godsill, S. J. (2003). Bayesian harmonic models for musical signal analysis. In *Bayesian Statistics 7*.
- de Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, pp. 1917–1930.
- de la Cuadra, P., Master, A., & Sapp, C. (2001). Efficient pitch detection techniques for interactive music. In *Proceedings of the 2001 International Computer Music Conference*, La Habana, Cuba.
- Desain, P., Aarts, R., Cemgil, A. T., Kappen, B., van Thienen, H., & Trilsbeek, P. (1999). Robust time-quantization for music. In *Preprint of the AES 106th Convention*, p. 4905(H4), Munich, Germany. AES.
- Desain, P., & Honing, H. (1991). Quantization of musical time: a connectionist approach. In Todd, P. M., & Loy, D. G. (Eds.), *Music and Connectionism.*, pp. 150–167. MIT Press., Cambridge, Mass.
- Desain, P., & Honing, H. (1994). A brief introduction to beat induction. In *Proceedings of ICMC*, San Francisco.
- Desain, P., Honing, H., & de Rijk, K. (1992). The quantization of musical time: a connectionist approach. In *Music, Mind and Machine: Studies in Computer Music, Music Cognition and Artificial Intelligence*, pp. 59–78. Thesis Publishers, Amsterdam.
- Dixon, S., & Cambouropoulos, E. (2000). Beat tracking with musical knowledge. In Horn, W. (Ed.), *Proceedings of ECAI 2000 (14th European Conference on Artificial Intelligence)*, Amsterdam.
- Doucet, A., & Andrieu, C. (2001). Iterative algorithms for state estimation of jump Markov linear systems. *IEEE Trans. on Signal Processing*, 49(6), 1216–1227.
- Doucet, A., de Freitas, N., & Gordon, N. J. (Eds.). (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York.
- Doucet, A., de Freitas, N., Murphy, K., & Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Uncertainty in Artificial Intelligence*.
- Doucet, A., Godsill, S., & Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197–208.

- Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York.
- Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, New York.
- Ellis, D. P. W. (1996). *Prediction-Driven Computational Auditory Scene Analysis*. Ph.D. thesis, MIT, Dept. of Electrical Engineering and Computer Science, Cambridge MA.
- Fearnhead, P. (2003). Exact and efficient bayesian inference for multiple changepoint problems. Tech. rep., Dept. of Math. and Stat., Lancaster University.
- Fletcher, N. H., & Rossing, T. (1998). *The Physics of Musical Instruments*. Springer.
- Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11.
- Ghahramani, Z., & Beal, M. (2000). Propagation algorithms for variational Bayesian learning. In *Neural Information Processing Systems 13*.
- Ghahramani, Z., & Hinton, G. (1998). Variational learning for switching state-space models. *Neural Computation*, 12(4), 963–996.
- Ghahramani, Z., & Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. (crg-tr-96-2). Tech. rep., University of Totronto. Dept. of Computer Science.
- Godsill, S., Doucet, A., & West, M. (2001). Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics*, 52(1), 82–96.
- Godsill, S. J., & Rayner, P. J. W. (1998). *Digital Audio Restoration - A Statistical Model-Based Approach*. Springer-Verlag.
- Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings Part F, Radar and Signal Processing*, Vol. 140(2), pp. 107–113.
- Goto, M., & Muraoka, Y. (1998). Music understanding at the beat level: Real-time beat tracking for audio signals. In Rosenthal, D. F., & Okuno, H. G. (Eds.), *Computational Auditory Scene Analysis*.
- Grubb, L. (1998). *A Probabilistic Method for Tracking a Vocalist*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- Hamanaka, M., Goto, M., Asoh, H., & Otsu, N. (2001). A learning-based quantization: Estimation of onset times in a musical score. In *Proceedings of the 5th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2001)*, Vol. X, pp. 374–379.
- Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge Univ. Press, Cambridge, U.K.
- Heijink, H., Desain, P., & Honing, H. (2000). Make me a match: An evaluation of different approaches to score-performance matching. *Computer Music Journal*, 24(1), 43–56.
- Heskes, T. Zoeter, O. (2002). Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings UAI*.
- Hess, W. J. (1983). *Pitch Determination of Speech Signal*. Springer, New York.
- Honing, H. (1990). Poco: An environment for analysing, modifying, and generating expression in music.. In *Proceedings of ICMC*, pp. 364–368, San Francisco.
- Hu, G., & Wang, D. (2001). Monaural speech segregation based on pitch tracking and amplitude modulation. Tech. rep. OSU-CISRC-12/01-TR25, Ohio State University.

- Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.
- Irizarry, R. A. (2001). Local harmonic estimation in musical sound signals. *Journal of the American Statistical Association*, to appear.
- Irizarry, R. A. (2002). Weighted estimation of harmonic components in a musical sound signal. *Journal of Time Series Analysis*, 23.
- Isard, M., & Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. In *ECCV (1)*, pp. 343–356.
- Jang, G. J., & Lee, T. W. (2002). A probabilistic approach to single channel blind signal separation. In *Neural Information Processing Systems, NIPS*2002*, Vancouver.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, 35–45.
- Kashino, K., Nakadai, K., Kinoshita, T., & Tanaka, H. (1995). Application of bayesian probability network to music scene analysis. In *Proc. IJCAI Workshop on CASA*, pp. 52–59, Montreal.
- Klapuri, A., Virtanen, ., Eronen, ., & Seppänen, . (2001). Automatic transcription of musical recordings. In *CRAC-01, Consistent and Reliable Acoustic Cues Workshop*, Aalborg, Denmark.
- Klapuri, A., Virtanen, T., & Holm, J.-M. (2000). Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals. In *COST-G6, Conference on Digital Audio Effects*.
- Klassner, F., Lesser, V., & Nawab, S. (1998). The IPUS blackboard architecture as a framework for computational auditory scene analysis. In Rosenthal/Okuno (Ed.), *Computational Auditory Scene Analysis*. Lawrence Erlbaum Inc.
- Kronland-Martinet, R. (1988). The wavelet transform for analysis, synthesis and processing of speech and music sounds. *Computer Music Journal*, 12(4), 11–17.
- Lane, . (1990). Pitch detection using a tunable IIR filter. *Computer Music Journal*, pp. 46–?
- Large, E. W., & Jones, M. R. (1999). The dynamics of attending: How we track time-varying events. *Psychological Review*, 106, 119–159.
- Lerner, U., & Parr, R. (2001). Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proc. UAI-01*, pp. 310–318, Seattle, Washington.
- Liu, J. S., Chen, R., & Logvinenko, T. (2001). A theoretical framework for sequential importance sampling with resampling. In Doucet, A., de Freitas, N., & Gordon, N. J. (Eds.), *Sequential Monte Carlo Methods in Practice*, pp. 225–246. Springer Verlag.
- Longuet-Higgins, H. C., & Lee, C. (1982). Perception of musical rhythms. *Perception*.
- Longuet-Higgins, H. C. (1987). *Mental Processes: Studies in Cognitive Science*. MIT Press, Cambridge. 424p.
- Longuet-Higgins, H. (1976). The perception of melodies. *Nature*, 263, 646–653.
- MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.
- Maher, R. C. (1990). Evaluation of a method for separating digitized duet signals. *Journal of the Audio Engineering Society*, 38(12), 956–979.
- Mani, R. (1999). Knowledge-based processing of multicomponent signals in a musical application. *Signal Processing*, 74(1), 47–69.

- Marthi, B., Pasula, H., Russell, S., & Peres, Y. (2002). Decayed MCMC filtering. In *Proceedings of UAI*.
- Martin, K. (1999). *Sound-Source recognition*. Ph.D. thesis, MIT.
- McAulay, R. J., & Quatieri, T. F. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4), 744–754.
- Mellinger, D. K. (1991). *Event Formation and Separation in Musical Sound*. Ph.D. thesis, Stanford University Dept. of Computer Science.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087–1091.
- Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Assoc.*, 44(247), 335–341.
- Michon, J. (1967). *Timing in Temporal Tracking*. Soesterberg: RVO TNO.
- Minka, T. (2001). *Expectation Propagation for approximate Bayesian inference*. Ph.D. thesis, MIT.
- Moorer, J. (1977). On the transcription of musical sound by computer. *Computer Music Journal*, 1(4), 32–38.
- Murphy, K. P. (1998). Switching Kalman filters. Tech. rep., Dept. of Computer Science, University of California, Berkeley.
- Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California, Berkeley.
- Olshausen, B., & Millman, J. (2000). Learning sparse codes with a mixture-of-Gaussians prior. In *NIPS*, Vol. 12, pp. 841–847. MIT Press.
- Parra, L., & Jain, U. (2001). Approximate Kalman filtering for the harmonic plus noise model. In *Proc. of IEEE WASPAA*, New Paltz.
- Piszczałski, M., & Galler, B. (1977). Automatic music transcription. *Computer Music Journal*, 1(3), 24–31.
- Plumbley, M., Abdallah, S., Bello, J. P., Davies, M., Monti, G., & Sandler, M. (2002). Automatic music transcription and audio source separation. *Cybernetics and Systems*, 33(6), 603–627.
- Pressing, J., & Lawrence, P. (1993). Transcribe: A comprehensive autotranscription program.. In *Proceedings of the International Computer Music Conference*, pp. 343–345, Tokyo. Computer Music Association.
- Qi, Y., Minka, T. P., & Picard, R. W. (2002). Bayesian spectrum estimation of unevenly sampled nonstationary data. Tech. rep. Vismod-TR-556, MIT Media Lab.
- Quinn, B. G., & Hannan, E. J. (2001). *The Estimation and Tracking of Frequency*. Cambridge University Press.
- Rabiner, L. R. (1989). A tutorial in hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), 257–286.
- Rabiner, L. R., Chen, M. J., Rosenberg, A. E., & McGonegal, A. (1976). A comparative study of several pitch detection algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23, 552–557.
- Raphael, C. (2001a). A mixed graphical model for rhythmic parsing. In *Proc. of 17th Conf. on Uncertainty in Artif. Int.* Morgan Kaufmann.

- Raphael, C. (2001b). A probabilistic expert system for automatic musical accompaniment. *Journal of Computational and Graphical Statistics*, 10(3), 467–512.
- Raphael, C. (2002). Automatic transcription of piano music. In *Proc. ISMIR*.
- Raphael, C., & Stoddard, J. (2003). Harmonic analysis with probabilistic graphical models. In *Proc. ISMIR*.
- Raphael, C. (2002). Automatic transcription of piano music. In *Proceedings of the International Symposium on Music Information Retrieval, IRCAM/Paris*.
- Rioul, O., & Vetterli, M. (1991). Wavelets and signal processing. *IEEE Signal Processing Magazine*, October, 14–38.
- Roberts, G. O., & Rosenthal, J. S. (1998). Markov Chain Monte Carlo: Some practical implications of theoretical results. *Canadian Journal of Statistics*, 26, 5–31.
- Rodet, X. (1998). Musical sound signals analysis/synthesis: Sinusoidal + residual and elementary waveform models. *Applied Signal Processing*.
- Rossi, L., Girolami, G., & Leca, M. (1997). Identification of polyphonic piano signals. *Acustica*, 83(6), 1077–1084.
- Rowe, R. (2001). *Machine Musichanship*. MIT Press.
- Roweis, S. (2001). One microphone source separation. In *Neural Information Processing Systems, NIPS*2000*.
- Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural Computation*, 11(2), 305–345.
- Saul, K. L., Lee, D. D., Isbell, C. L., & LeCun, Y. (2002). Real time voice processing with audiovisual feedback: toward autonomous agents with perfect pitch. In *Neural Information Processing Systems, NIPS*2002*, Vancouver.
- Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of Acoustical Society of America*, 103:1, 588–601.
- Scheirer, E. D. (2000). *Music-Listening Systems*. Ph.D. thesis, Massachusetts Institute of Technology.
- Serra, X., & Smith, J. O. (1991). Spectral modeling synthesis: A sound analysis/synthesis system based on deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4), 12–24.
- Sethares, W. A., & Staley, T. W. (1999). Periodicity transforms. *IEEE Transactions on Signal Processing*, 47(11), 2953–2964.
- Shumway, R. H., & Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the em algorithm. *J. Time Series Analysis*, 3(4), 253–264.
- Slaney, M. (1995). Pattern playback from 1950 to 1995. In *Proceedings of 1995 IEEE Systems, Man and Cybernetics Conference*, Vancouver, Canada. IEEE.
- Smith, J. O. (1992). Physical modeling using digital waveguides. *Computer Music Journal*, 16(4), 74–87.
- Smith, L. (1999). *A Multiresolution Time-Frequency Analysis and Interpretation of Musical Rhythm*. Ph.D. thesis, University of Western Australia.
- Smyth, P., Heckerman, D., & Jordan, M. I. (1996). Probabilistic independence networks for hidden markov probability models MSR-TR-96-03. Tech. rep., Microsoft Research.

- Sterian, A. (1999). *Model-Based Segmentation of Time-Frequency Images for Musical Transcription*. Ph.D. thesis, University of Michigan, Ann Arbor.
- Tanizaki, H. (2001). Nonlinear and non-Gaussian state-space modeling with Monte Carlo techniques: A survey and comparative study. In Rao, C., & Shanbhag, D. (Eds.), *Handbook of Statistics, Vol.21: Stochastic Processes: Modeling and Simulation*. North-Holland.
- Temperley, D. (2001). *The Cognition of Basic Musical Structures*. MIT Press.
- Terhardt, E. (1974). Pitch, consonance and harmony. *Journal of the Acoustical Society of America*, 55(5), 1061–1069.
- Thom, B. (2000). Unsupervised learning and interactive jazz/blues improvisation. In *Proceedings of the AAAI2000*. AAAI Press.
- Todd, N. P. M. (1994). The auditory “primal sketch”: A multiscale model of rhythmic grouping. *Journal of new music Research*.
- Toiviainen, P. (1999). An interactive midi accompanist. *Computer Music Journal*, 22:4, 63–75.
- Truong-Van, B. (1990). A new approach to frequency analysis with amplified harmonics. *J. Royal Statistics Society B*, pp. 203–222.
- Tzanetakis, G. (2002). *Manipulation, Analysis and Retrieval Systems for Audio Signals*. Ph.D. thesis, Princeton University.
- Valimaki, V., Huopaniemi, J., Karjalainen, & Janosy, Z. (1996). Physical modeling of plucked string instruments with application to real-time sound synthesis. *J. Audio Eng. Society*, 44(5), 331–353.
- Vercoe, B. (1984). The synthetic performer in the context of live performance. In *Proceedings of ICMC*, pp. 199–200, San Francisco. International Computer Music Association.
- Vercoe, B., & Puckette, M. (1985). The synthetic rehearsal: Training the synthetic performer. In *Proceedings of ICMC*, pp. 275–278, San Francisco. International Computer Music Association.
- Vercoe, B. L., Gardner, W. G., & Scheirer, E. D. (1998). Structured audio: Creation, transmission, and rendering of parametric sound representations. *Proc. IEEE*, 86:5, 922–940.
- Walmsley, P. J. (2000). *Signal Separation of Musical Instruments*. Ph.D. thesis, University of Cambridge.
- Weintraub, M. (1985). *A Theory and Computational Model of Auditory Monaural Sound Separation*. Ph.D. thesis, Stanford University Dept. of Electrical Engineering.
- West, M., & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models* (Second edition). Springer-Verlag.

Samenvatting

Muziektranscriptie kan worden beschouwd als het omzetten van een gedigitaliseerde opname van een muziekuitsvoering in een beschrijving die voor mensen te lezen en te begrijpen is. Het einddoel van onderzoek in deze richting is een computer programma te ontwerpen dat uit iedere voorstelbare uitvoering een muzikale beschrijving (b.v. in de gebruikelijke westelijke muzieknotatie) kan afleiden, die onder andere de toonhoogten en posities van noten op een bepaald tijdstip bevat. Transcriptie van iedere willekeurige muziekuitsvoering zonder enige aanname van de soort muziek is een zeer moeilijke, zelfs “AI-complete” taak, waarvoor men menselijke intelligentie zou moeten kunnen reproduceren. Wij geloven echter dat onder sommige realistische aannames een werkbare oplossing kan worden gevonden; namelijk door een combinatie van kennis vanuit verschillende wetenschappen, zoals de cognitieve wetenschappen, muziekwetenschap en akoestiek, en computationele technieken afkomstig uit de kunstmatige intelligentie, automatisch leren en digitale signaal verwerking. Het doel van dit proefschrift is om een praktische aanpak voor muziektranscriptie te ontwikkelen door deze grote hoeveelheid a-priori kennis in een samenhangend en transparent computationeel model te integreren.

In dit proefschrift behandelen we muziektranscriptie als een statistisch inferentie probleem, waarbij we een notatie zoeken die een gegeven muziek signaal goed beschrijft. In deze context identificeren we drie problemen, namelijk *ritme kwantisatie*, *het volgen van tempo* en *van polyfone pitch*. Voor elk probleem definiëren we een generatief kansmodel. De transcriptie taak is dan gedefinieerd als het “omdraaien” van dit generatief model om zo de originele “verborgen” notatie te vinden.

In hoofdstuk 2 definiëren we een kansmodel tussen korte notaties en hun waargenomen uitvoering. Uit psychoakoestische experimenten blijkt dat zelfs voor vrij eenvoudige ritmes getrainde transcriptie experts verschillende antwoorden kunnen geven. We laten zien hoe een kansmodel deze onzekerheid op een natuurlijke manier kan vatten en leren.

In hoofdstuk 3 ligt onze aandacht op volgen van tempo variaties. In dit model wordt het tempo gezien als een verborgen variabele die we door Kalman filtering schatten.

De volgende hoofdstuk (hoofdstuk 4) introduceert een generatief model voor ritme kwantisatie en tempo volgen tegelijkertijd. Het kansmodel is een zogenaamd “switching state space” model. In dit model is het niet mogelijk om kansen exact te berekenen, daarom behandelen we hier benaderingsmethoden als Markov Chain Monte Carlo (MCMC) en sequential Monte Carlo (SMC).

In de laatste hoofdstuk 5 beschrijven we een model voor polyfone transcriptie vanuit een audio signaal. Het model, uitgedrukt als een “Dynamic Bayesian Network” (dynamisch Bayesiaans Netwerk), bevat de afhankelijkheid tussen het signaal en een piano rol. Dit model is ook een speciaal geval van het switching state space model. Waar mogelijk leiden we polynomiale tijd algoritmen af en anders effectieve benaderingsmethoden.

De meest aantrekkelijke eigenschap van de Bayesiaanse aanpak voor muziektranscriptie is ont koppeling van het model en het benaderingsalgoritme. In dit raamwerk beschrijft het model duidelijk het doel maar de vraag hoe dit doel te bereiken, hoewel zeer belangrijk, wordt een onafhankelijke kwestie. In perceptuele taken en in muziektranscriptie in het bijzonder is de vraag “wat te optimaliseren” niet eenvoudig te beantwoorden. Dit proefschrift probeert een antwoord

te geven door doelfuncties te definiëren geschreven als probabilistische grafische modellen en introduceert benaderende en exacte inferentie technieken om een acceptabele oplossing efficiënt te vinden.

translated and edited with Matthijs Spaan

Dankwoord

Sometime around November of 97, I was randomly browsing through the internet and totally coincidentally, ended up reading archives of a mailing list. A job advertisement on music transcription draw my attention and although the application deadline has already passed, (deadlines are made for passing, right?), I decided to send in my CV. To my surprise, I was invited to give a talk and even was offered the job! I would be working as a computer scientist in a music related project at a biophysics department located in a medical faculty with collaboration of researchers from psychology. Although, frankly speaking, that wasn't exactly the setup I was expecting, I accepted the kind offer without much hesitation.

The start of the research described in this thesis may be marked by this occasion, but it would never have been finished without collaboration, support and input from many others. From the beginning until the end, Bert has been always there both as a friend and as an advisor in shaping the research direction while giving me enough freedom to explore. This thesis would never have reached its level without his encouragement, support, critical attitude and his enormous expertise.

Perhaps I was the most lucky because David was a postdoc at SNN when I was starting. Talking to him has been (and still is) a constant source of inspiration and learning experience. I am very grateful to the other members of SNN (past and present), Wim, Tom, Machiel (*ne haber abi*), Onno, Martijn and Alex; I have always benefited from their constructive critics, impromptu discussions and inspiration. I am also grateful to Stan for creating and sustaining such a great work environment to make all this happen and Annet for her support in providing shelter, food, transportation; practically solving every "real-life" problem but most of all for her friendship.

Without the collaboration of Peter and Henkjan, I would have been most probably ignorant about many music perception issues and it would have not been possible to test the models on systematically collected data. Therefore, I would like to thank them and other members of the Music, Mind and Machine group: Paul, Rinus, Chris and Huub. I am indebted to STW for providing the funding for my research and extensive worldwide travel. Without this, I would not have been able to pursue my research with the freedom I enjoyed. I also would like to thank Ben Kröse for letting me finish up in Amsterdam, and other members of the Amsterdam crowd, especially to Nikos, who has provided his support and feedback in many ways and to Matthijs; who was very helpful in the Dutch translation of the summary.

And last, but not the least, I wish to thank my office-mate, the other member of the holy "tea-room" brotherhood: Niels "Sonny-Rollins" Cornelisse. He has been the primary victim of the collateral damage caused by my tempo tracking experiments but I have always enjoyed our endless jazz-chats and the sco-coltrane-jamesbrown-chrispotter-jaco mix we have been broadcasting to the corridor. Many thanks to Marjan, Noel, Vivek, Sonja, Marcel, Paul, Anton, Thom and everyone else in the lab who have made it a fun place to be. I want to mention and thank the support group, Günter and Ger who made sure everything works properly, and Hans for helping in fixing up broken things.

I am deeply thankful to my former professors from Boğaziçi University, in particular to Alp Eden for a short but wonderful journey through the "ivory castle" of mathematics and Ethem Alpaydın for introducing and teaching me to the basic methods of machine learning. My special

thanks go to Khalid Sayood not only for being a wonderful teacher in data compression, signal processing and information theory but most of all for his attitude and great spirit that has been very influential on me. During this period, I had also the very good fortune to have Cüneyt as a friend and a college to exchange ideas. I had also the privillage to have great friends, who happened to be also “natural born experts” in various fields of music from whom I learned a lot : Alper Maral, Çağlayan Öge, Cengiz Baysal, Sarp Maden, Tuluğ Tırpan, Tolga Tüzün, Selim Benba and Cumhur Erkut.

Finally, I want to thank Çağlayan, Selma, Oğuz, İlkur, the members of our small but fully connected Amsterdam clique. Without your music, support and friendship, it would have been impossible to maintain a decent level of sanity throughtout the difficult times.

To express my gratitute to my family, I have to switch to my native language; although I surely believe it would have been hard to express this in any language.

Annem ve Babama; Anneanneme; Babaannem ve Dedeme, hayatta bir şeyler başarabildiysen sizin omuzlarınızda durduğum içindir.

Aslı’cığıma, her zaman yanımda olduğunu ve olacağını hissettirdiği için, (Aslı’cım, ...)

Nazife (1912-2003) ve Adnan (1909-2001) Cemgil’in anısına

Curriculum Vitae

Ali Taylan Cemgil was born 28 January 1970, in Ankara, Turkey. He grew up in Istanbul and obtained his secondary education from Istanbul Erkek Lisesi. After highschool, he entered Boğaziçi University, Istanbul, to study computer engineering. During the university years, parallel to his studies, he started to perform in local clubs as a jazz musician (electric bass). During the last year of his undergraduate studies, he became interested in computer music, signal processing and artificial intelligence. After graduating from the university in 1993, besides working as a professional musician, he worked for a year as a programmer but eventually decided to attend the graduate studies program in Computer Engineering at Boğaziçi University, where he would be employed as a teaching assistant until 1998. He obtained a masters degree in 1995 where he has investigated sub-band coding techniques (wavelets) for music transcription. The same year, he enrolled to the PhD program and decided to focus on probabilistic models for music analysis. In 1998, he moved to the Netherlands and joined SNN, Nijmegen University as a Ph.D. student under supervision of Bert Kappen to work on Bayesian methods and their application to music transcription. Between 2001 and 2004, he also studied at the jazz department of the Amsterdam Conservatory. Since 2003, he is working as a researcher at Intelligent Autonomous Systems group at University of Amsterdam.