Markov-switching State Space Models for Uncovering Musical Interpretation

Daniel J. McDonald*
Department of Statistics, Indiana University
and
Michael McBride
Department of Statistics, Indiana University

March 4, 2019

Abstract

For concertgoers, musical interpretation is the most important factor in determining whether or not we enjoy a classical performance. Every performance includes mistakes—intonation issues, a lost note, an unpleasant sound—but these are all easily forgotten (or unnoticed) when a performer engages her audience, imbuing a piece with novel emotional content beyond the vague instructions inscribed on the printed page. While music teachers use imagery or heuristic guidelines to motivate interpretive decisions, combining these vague instructions to create a convincing performance remains the domain of the performer, subject to the whims of the moment, technical fluency, and taste. In this research, we use data from the CHARM Mazurka Project—forty-six professional recordings of Chopin's Mazurka Op. 63 No. 3 by consumate artists—with the goal of elucidating musically interpretable performance decisions. Using information on the tempo the recordings, we apply functional data analysis techniques enriched with prior information gained from music theory to discover relevant features and perform hierarchical clustering. The resulting clusters suggest methods for informing music instruction, discovering listening preferences, and analyzing performances.

Keywords: keyword1; keyword2;

*The authors gratefully acknowledge support from the National Science Foundation (grants DMS-1407439 and DMS-1753171).

1 Introduction

Note: See here for the style guide (detailed) and here for the author instructions.

In recent years, statistical analysis of recorded music has become more and more important to academics and industry. Online music services like Pandora, Last.fm, Spotify, and others rely on recommendation systems to suggest potentially interesting or related songs to listeners. In 2011, the KDD Cup challenged academic computer scientists and statisticians to identify user tastes in music with the Yahoo! Million Song Dataset (see Dror et al. (2012) for details of the competition). Pandora, through its proprietary Music Genome Project, uses trained musicologists to assign new songs a vector of trait expressions (consisting of up to 500 'genes' depending on the genre) which can then be used to measure similarity with other songs. However, most of this work has focused on the analysis of more popular and more profitable genres of music—pop, rock, country—as opposed to classical music.

Western classical music, classical music for short, is a subcategory of music whose boundaries are occasionally difficult to define. But the distinction is of great importance when it comes to the analysis which we undertake here. Leonard Bernstein, the great composer, conductor and pianist, gave the following characterization in one of his famous "Young People's Concerts" broadcast by the Columbia Broadcasting Corporation in the 1950s and 1960s (Bernstein, 2005).

> You see, everybody thinks he knows what classical music is: just any music that isn't jazz, like a Stan Kenton arrangement or a popular song, like "I Can't Give You Anything but Love Baby," or folk music, like an African war dance, or "Twinkle, Twinkle Little Star." But that isn't what classical music means at all.

Bernstein goes on to discuss an important distinction between what we often call 'classical music' and other types of music which is highly relevant to the current study.

> The real difference is that when a composer writes a piece of what's usually called classical music, he puts down the exact notes that he wants, the exact instruments or voices that he wants to play or sing those notes—even the exact number of instruments or voices; and he also writes down as many directions as he can think of. [...] Of course, no performance can be perfectly exact, because

Figure 1: The tempo (beats/minute) of a 2003 recording attributed to Joyce Hatto.

> there aren't enough words in the world to tell the performers everything they have to know about what the composer wanted. But that's just what makes the performer's job so exciting—to try and find out from what the composer did write down as exactly as possible what he meant. Now of course, performers are all only human, and so they always figure it out a little differently from one another.

What separates classical music from other types of music is that the music itself is written down but performed millions of times in a variety of interpretations. There is no 'gold standard' recording to which everyone can refer, but rather a document created for reference. Therefore, the musical genome technique mentioned above will serve only to relate 'pieces' but not 'performances'. We need new methods in order to decide whether we prefer Leonard Bernstein's recording of Beethoven's Fifth Symphony or Herbert von Karajan's and to articulate why.

Musical recordings are complex data files that describe the intensity and onset time for every keystroke made by the performer. Matching this data to a musical score, removing incorrect notes, anticipating note onsets for automated accompaniment, comparing diverse performances, and discovering the relationship between performer choice and listener enjoyment all require "smoothing" the performance data so as to find low-dimensional structure. Statistical techniques like smoothing splines presume small changes in a derivative. But musical performances do not conform to these assumptions because tempo and dynamic interpretations rely on the juxtaposition of local smoothness with sudden changes and emphases to create listener interest. It is exactly the parts of a performance that are poorly described by statistical smoothers that render a performance interesting. Furthermore, many of these inflections are notated by the composer or are implicit in performance practice developed over centuries of musical expressivity. Consequently, regularization that incorporates domain knowledge leads to better statistical and empirical results (McDonald and McBride, 2018).

Figure 1 shows (blue dots) the note-by-note tempo of a 2003 recording attributed to Joyce Hatto. Splines with equally spaced knots (orange/dotted) are too smooth, and choosing locations to duplicate knots manually (red/dashed) to coincide with musical phrase endings works better. The solid green line shows a learned musical pattern from a Markov Switching state-space model we developed which can automatically learn tempo emphases (for example, near measure 40), where the performer plays individual notes slightly slower than the prevailing tempo, and automatically discover phrases without purposeful knot duplication. Interestingly, such musical analyses can help to compare performances—it was discovered in 2006 that this particular recording was actually made in 1988 by Eugen

1.1 Related work

One of the biggest reasons for having an off-line score-to-MIDI alignment is to create data sets for quantitatively studying music performance, which is a research area that receives much attention. To name a few, N.P. Todd has a series of works on computational modeling of timing and dynamics [Tod85] [Tod89] [Tod92], B.H. Repp has a series of research efforts on comparing different performances of the same music from different performers [Rep90] [Rep92] [Rep95]. S. Flossmann and G. Widmer have a series of studies on machine learning and rendering expressive performances [WFG09] [FGG+10] [FW11] [FGW13]. We also have a series of studies on performance interpretation parsing [GR12] [GR13]. All of these studies need data sets of MIDI performances with ground truth.

The existing MIDI data sets with ground truth are limited and some are copy- righted [FGG+10]. Also, none of these data are create solely with a score-performance alignment program because automated score-performance often contain many errors [FW11]. Researchers need data sets to explore the alignment algorithms. But without a good alignment algorithm it is often very hard to create enough data sets.

Online alignment: As opposed to the off-line version, on-line alignment doesn't have access to "future data". It processes performance actions in real-time as they are acquired. This version of alignment is often referred to as score following. [OLS03] has an annotated bibliography detailing the works of score following. Some of the recent developments can be found at [RG09] [Con10] [NTS14].

One of the most direct applications of score following is automatic music accom- paniment. Active research on this topic continued for over two decades since the simultaneous premier of the first two such systems at the ICMC in 1984 [Dan84] [Ver84]. These systems seek to provide a flexible accompaniment to a live soloist that follows expressive timing and other performance nuances exhibited by the soloist.

While there are some impressive successes for monophonic instruments in highly challenging domains [OLS03] [Rap04], a reliable accompaniment system for classical piano concerto for acoustic piano signal still pose challenges due to the high polyphony nature of the music [SOS04]. Some variant of HMM is used in recent development of audio score following for piano. But such research still calls for further development [CC14].

Frequency of errors in piano: Even with highly skilled pianists, these errors occur far more often than one may expect. There are studies in which the error rate could go up to 10%, even for highly skilled pianists [FGG+10]. Section 4.3.1 discusses the error rate in detail.

Expressive timing: Although most approaches in the literature mainly focus on using the pitch informa- tion while dealing with notes in chronological order [Dan84] [PL92] [BBZ93] [Lar93], it has been shown that the IOIs between note clusters can also be useful in solving the alignment problem [Van95] [GM11]. Expressive timing is the deviation of inter- onset intervals from the written score [PVdS93]. It is one of the most important contributions that musicians give to bring music to life.

[Ros92b] conducted an experiment on rhythmic tolerance. He observed that the listeners are expecting the IOIs to be close to their nominal lengths using the tempo marking, and the IOIs with same nominal length have an asymmetric distribution. Based on these observations, [Van95] proposes an online alignment system that mainly uses timing information. In this system, several recent IOIs are stored to compute a local average tempo. If the next IOI falls within a range determined by the average tempo, the system makes a match. A pitch-matching algorithm is used only when a match cannot be found. Despite its simplicity, this system shows its robustness when incorrect pitches are played at the expected time. The author shows reasonable results without using much pitch information. This can be seen as a strong indication that the timing information could be useful in a score-alignment system. [GM11] adopted the idea of using a local tempo model in an off-line matching algorithm. In this multi-pass alignment algorithm, unexpected IOIs computed from the local tempo are used to identify inserted notes.

Modeling tempo: Although the efforts in music modeling using quantitative methods have been increas- ing over the past two decades [Tod89] [DH94] [WG04] [GW11] [GK14], there are still very few modeling assumption that can be translated into familiar musical meaning. For example, we may be able to get the loudness of every note played by a pianist, but it is different from the term dynamic used by musicians, which is often referred to as a loudness trend over a group of notes. Similarly, when musicians talk about tempo, they usually mean changes happening over a period of time rather than the something related to the time difference of two consecutive notes they played.

Utility of having a model: One of the most straightforward applications is performance visualization. Music is communicated through sound. From the hearing point of view, the information we have direct access to at any given time is very limited – the sound we are hearing at the moment, which derives its meaning from context. Thus, it is time consuming for us to "browse" a performance using our ears since we have to listen as the music plays. Visualization is a tool to transform the sound into an image so that we can utilize our eyes to explore information within a certain time period at once. It opens a whole new world to describe, analyze and compare music. Also, visualization is often an interesting and rewarding experience for musicians while reviewing their performances. It provides a very different perspective for musicians to see what they have done or compare side by side with other performance of the same music.

Most music visualization research focuses on analyzing music structure such as pattern and repetition [LNS07] [PK08] [WB10]. These visualization systems research and identify the structure of music. They provide tools for listeners to navigate through music quickly.

In the area of creating music, musicians have a long standing interest in improving and creating performances that aim towards perfection. Before the appearance of recording, the only way to improve a performance is to play it again. If someone wants a perfect performance, it has to be played perfectly – from the first note to the last note. With the development of recording technology, it became possible to splice sections of performances together. If there is an unsatisfactory part in a recording, it can be replaced. Thus, a performance can be improved and perfected section by section. But the convenience comes with a price: it is often not easy to make the splicing inaudible. Subtle things such as the sound change caused by different humidity levels, the slight inconsistency of articulation, the different dynamics from different takes and the different tempo variation from different takes could affect the quality of splicing. Also, the whole work flow is very labor intensive and can only be done by trial and error. But to this day, splicing is still one of the most commonly used techniques for improving recordings of performances.

With the development of computer technology, there is a growing interest in gen- erating performances that can match the level of a trained musician. Most existing rendering systems are rule-based or case-based. Such systems often include extracting and applying rules with preset parameters [SUZ03] [HBHK04] [FBS06]. The weak- ness of rule-based or case-based systems is that it is still debatable whether we can find a set of rules/cases that can cover what it takes to make a meaningful music. There are also several statistical approaches [FW11] [WFG09]. These approaches use statistical methods to train a note by note performance model from a large quantity of data (e.g. a complete recording of Chopin piano works from a reproducing piano). However, compared to the heavy parametrization of these models, the data sets are not as big as they looks. Thus, overfitting could occur in these models.

An accompaniment system can benefit from such a low-dimensional representation too. A traditional accompaniment system seeks to create a flexible accompaniment to a live soloist that follows the player [Dan84] [Rap03] [CEGJ12]. Most existing systems use the same following strategy to keep up with the soloist throughout a single piece. This could inevitably result in overfitting the soloist's performance and failing to understand the player's real intentions. Good following requires a deeper understanding of the performers' intention, thus separating signal from noise. A low-dimensional representation provides a higher-level view of the music, which has the potential to recognize the different musical characters in different sections of a piece (e.g. the tempo of a section recognized as ritardando is expected to slow down gradually while the tempo is not expected to vary a lot in a section recognized as steady tempo). Different following strategies can be adapted to better follow performance within sections provided by such a representation.

- We want to model tempo and dynamic decisions.

- We want a musician to understand what the parameters mean.

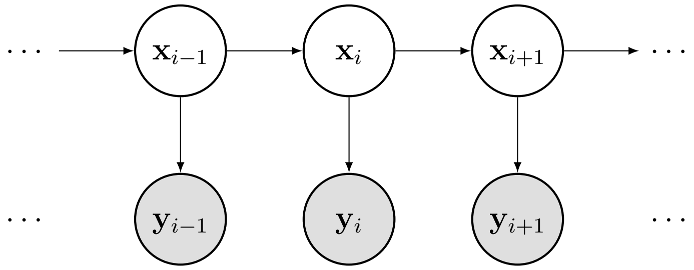

Figure 2: State-space model. Filled objects are observed, circles indicate that both hidden and observed states are continuous.

2 Materials and methods

2.1 Data and preprocessing

2.2 Switching state-space models

State-space models define the probability distribution of a time series Y by reference to some imagined, hidden state, X. In particular, the observation at a particular time t is assumed to be independent of past and future observations conditional on the state at time t. Coupled with temporal dependence for X—most frequently obeying the Markov property—induces a temporal model for the observations. The most general form of a state-space model is then characterized by the observation equation (the conditional probability of observations given the states), the state transition equation (specifying the nature of Markovian dynamics), and an initial distribution for the state:

$$y_i = f_\theta(x_i, \epsilon_i), \quad x_{i+1} = g_\theta(x_i, \eta_i), \quad x_1 \sim F, \tag{1}$$

where ϵ_i are η_i are marginally independent and identically distributed (IID) as well as mutually independent. Both y_i and x_i can be (generally) vector-valued, though in our application, y_i will be univariate. The vector $\{y_i\}_{i=1}^n$ is observed, and the goal is to make inferences for the unobserved states $\{x_i\}_{i=1}^n$ as well as any unknown parameters θ characterizing f_θ, g_θ, and the distributions of ϵ_i and η_i. Figure 2 shows a directed acyclic graph for the dependence structure in the typical state-space model.

In the case where f_θ and g_θ are linear with ϵ_i and η_i normally distributed, (1) specializes

Algorithm 1 Kalman filter: estimate x_i conditional on $\{y_j\}_{j=1}^i$, for all $i = 1, \ldots, n$ and calculate the log likelihood for θ

```
**Input:** $Y$, $x_0$, $P_0$, $d$, $T$, $R$, $c$, $Z$, and $G$
$\ell(\theta) \leftarrow 0$                                              ▷ Initialize the log-likelihood
**for** $i = 1$ to $n$ **do**
    $H = RQR^\top$                                                       ▷ Effective state variance
    $\chi_i \leftarrow d + Tx_{i-1|i-1}, \quad P_i \leftarrow H + TP_{i-1|i-1}T^\top$      ▷ Predict current state
    $\widetilde{y}_i \leftarrow c + Z\chi_i, \quad F_i \leftarrow G + ZP_iZ^\top$            ▷ Predict current observation
    $v_i \leftarrow y_i - \widetilde{y}_i \quad K_i \leftarrow P_iZ^\top F^{-1}$             ▷ Forecast error and Kalman gain
    $x_{i|i} \leftarrow \chi_i + K_iv_i, \quad P_{i|i} \leftarrow P_i - P_iZ^\top K_i$      ▷ Update
    $\ell(\theta) = \ell(\theta) - v_i^\top F^{-1} v_i - \log(|F_i|)$
**end for**
**return** $\widetilde{Y} = \{\widetilde{y}_i\}_{i=1}^n,\ \chi = \{\chi_i\}_{i=1}^n,\ \widetilde{X} = \{x_{i|i}\}_{i=1}^n,\ P = \{P_i\}_{i=1}^n,\ \widetilde{P} = \{P_{i|i}\}_{i=1}^n,\ \ell(\theta)$
```

Algorithm 2 Kalman smoother (Rauch-Tung-Striebel): estimate $\widehat{X}$ conditional on Y

```
**Input:** $\chi$, $\widetilde{X}$, $P$, $\widetilde{P}$, $T$, $c$, $Z$.
$t = n$,
$\widehat{x}_n \leftarrow \widetilde{x}_n$,
**while** $t > 1$ **do**
    $\widehat{y}_i \leftarrow c + Z\widehat{x}_i$,                                   ▷ Predict observation vector
    $e \leftarrow \widehat{x}_i - \chi_i, \quad V \leftarrow P_i^{-1} \quad ,$
    $t \leftarrow i - 1$,                                                 ▷ Increment
    $\widehat{x}_i = \widetilde{x}_i + \widetilde{P}_i TVe$
**end while**
**return** $\widehat{Y} = \{\widehat{y}_i\}_{i=1}^n, \widehat{X} = \{\widehat{x}_i\}_{i=1}^n$
```

to

$$\begin{aligned} x_i &= d + Tx_i + R\eta_i, \quad \eta_i \sim N(0,\ Q), \quad x_1 \sim N(x_0,\ P_0), \\ y_i &= c + Zx_i + \epsilon_i, \qquad \epsilon_i \sim N(0,\ G), \end{aligned} \tag{2}$$

where the matrices d, T, R, c, Z, R, Q, and G are allowed to depend on θ and can potentially vary (deterministically) with i. In this case, the Kalman filter, Algorithm 1 (see e.g. Harvey, 1990; Kalman, 1960), can be used to derive closed form solutions for the conditional distributions of the states and to calculate the likelihood of θ given data.

While Algorithm 1 returns the likelihood for θ, χ_i and P_i represent the mean and variance of the conditional distribution of the unobserved component given only the observations $\{y_j\}_{j=1}^i$: $\chi_i = \mathbb{E}[x_i \mid y_1 \ldots, y_i]$ and $P_i = \mathbb{V}[x_i \mid y_1, \ldots, y_i]$. To incorporate all future observations into these estimates, the Kalman smoother is required. There are many different smoother algorithms tailored for different applications. Algorithm 2, due to Rauch et al. (1965), is often referred to as the classical fixed-interval smoother (Anderson and Moore, 1979). It produces only the unconditional expectations of the hidden state $\widehat{x}_i = \mathbb{E}[x_i \mid y_1, \ldots, y_n]$ for the sake of computational speed. This version is more appropriate for inference in the type of switching models we discuss below.

Linear Gaussian state-space models can be made quite flexible by expanding the state vector or allowing the parameter matrices to vary with time. Furthermore, this general form encompasses many standard time series models: ARIMA models, ARCH and GARCH models, stochastic volatility models, exponential smoothers, and more (see Durbin and Koopman, 2001, for many other examples). Nonlinear, non-Gaussian versions have been extensively studied (Durbin and Koopman, 1997; Fuh, 2006; Kitagawa, 1987, 1996) and algorithms for filtering, smoothing, and parameter estimation have been derived (e.g., Andrieu et al., 2010; Koyama et al., 2010). However, these models are less useful for change-point detection or other forms of discontinuous behavior when the times of discontinuity are unknown.

To remedy this deficiency, one can use a switching state-space model as shown in Figure 3. Here, we assume S is a hidden, discrete process with Markovian dynamics. Then, the value of the hidden state at time t, $s_i = k$ say, can determine the evolution of the continuous model at time t. The graphical model in Figure 3 gives the conditional independence properties we will use in our model for musical interpretation, but this represents just one of many possibilities. Switching state-space models have a long history with many applications from economics (Hamilton, 2011; Kim, 1994; Kim and Nelson, 1998) to speech processing (Fox et al., 2011) to animal movement (Block et al., 2011; Patterson et al., 2008). An excellent overview of the history, typography, and algorithmic developments can be found in (Ghahramani and Hinton, 2000). In (2), the parameter matrices were not time varying. In our switching model, we allow the switch states s_i, s_{i-1}, along with the parameter vector θ, to determine the specific dynamics at time t:

$$\begin{aligned}
x_1 &\sim N(x_0,\ P_0), \\
x_{i+1} &= d(s_i, s_{i-1}) + T(s_i, s_{i-1})x_i + R(s_i, s_{i-1})\eta_i, \quad && \eta_i \sim N(0, Q(s_i, s_{i-1})), \\
y_i &= c(s_i) + Z(s_i)x_i + \epsilon_i, && \epsilon_i \sim N(0, G(s_i)).
\end{aligned} \tag{3}$$

In other words, the hidden Markov (switch) state determines which parameter matrices

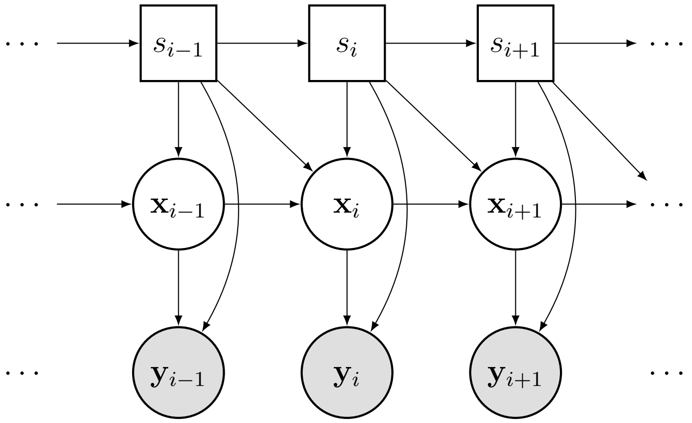

Figure 3: Switching state space model. Filled objects are observed, rectangles are discrete, and circles are continuous.

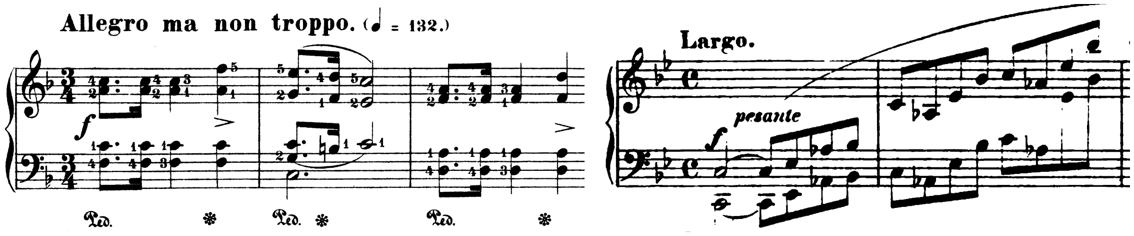

Figure 4: The beginning of two Chopin piano compositions: the Mazurka we analyze is on the left while the Ballade No. 1, Op. 23 is on the right.

govern the evolution of the system.

2.3 A model for tempo decisions

In musical scores, tempi (the Italian plural of *tempo*) may be marked at various points throughout a piece of music. The beginning can be either explicit, with a metronome marking to indicate the number of beats per minute (bpm), and/or with some words (e.g. Adagio, Presto, Langsam, Sprightly) which indicate an approximate speed. Figure 4 shows the beginning of two Chopin piano compositions: the Mazurka we analyze and the Ballade No. 1, Op. 23. The initial tempo of the Mazurka is given with a metronome marking as well as the Italian phrase *Allegro ma non troppo* ("cheerful, but not too much"). The beginning of the Ballade is marked *Largo*, which translates literally as "broad" or "wide", and modified by the stylistic indication *pesante* ("heavy"). Obviously, the metronome markings are much more exact, though even these are often viewed as suggestions rather than commandments.

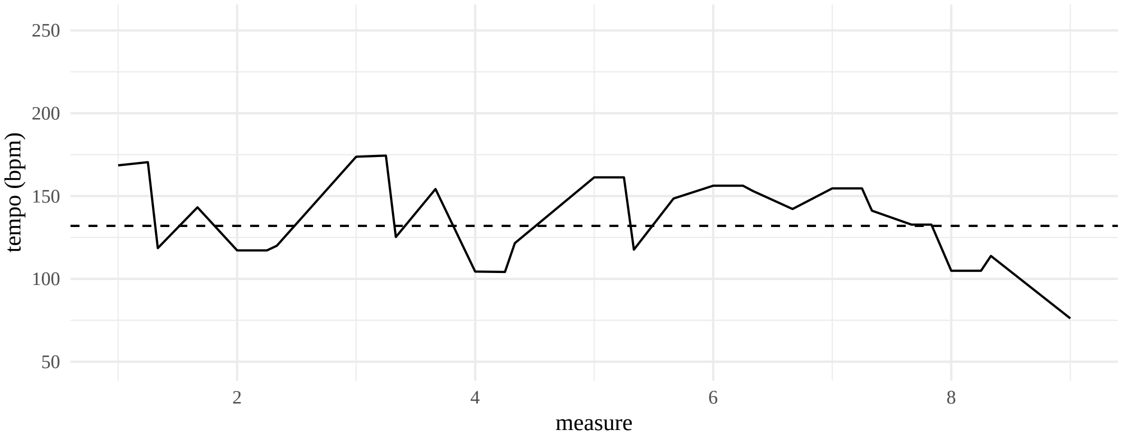

Figure 5: The solid line shows the observed note-by-note tempo for the beginning of the Mazurka as performed by Arthur Rubinstein in 1961. The dashed line indicates 132 bpm.

The metronome markings in most of Beethoven's compositions, for example, are notoriously fast, and some scholars believe that his metronome (one of the first ever made) was inaccurate (Forsén et al., 2013). Often, compositions will have numerous such markings later in the piece of music, but these are only some of the ways that tempo is indicated. Composers will also indicate periods of speeding-up (*accelerando*) or slowing-down (*ritardando*).

Absent instructions from the composer, performers generally maintain (or try to maintain) a steady tempo, and this assumption plays a major role in our model of tempo decisions. Of course, a normal human being never plays precisely like a metronome, although they may try quite hard to do so. The actual tempo is therefore best viewed as stochastic, the sum of an intentional, constant component, plus noise representing inaccuracy or, perhaps more charitably, unintentional variation which the listener fails to perceive as "wrong". For instance, the example in Figure 5 shows the beginning of the piece as performed by Arthur Rubinstein in a 1961 recording. The solid line shows the actual, performed tempo, while the dashed horizontal line is placed at the indicated tempo of 132 bpm. The figure has three important lessons: (1) actual tempo varies around intended tempo; (2) 132 bpm is not necessarily the tempo a performer will choose despite the indication; and (3) performers have other tempo intentions which are not marked, like the pronounced slow-down in measures 7–8.

Estimating intended *tempi* would be reasonably simple, perhaps, if the locations of the tempo changes were known. In such a case, the average of tempi between changes may be

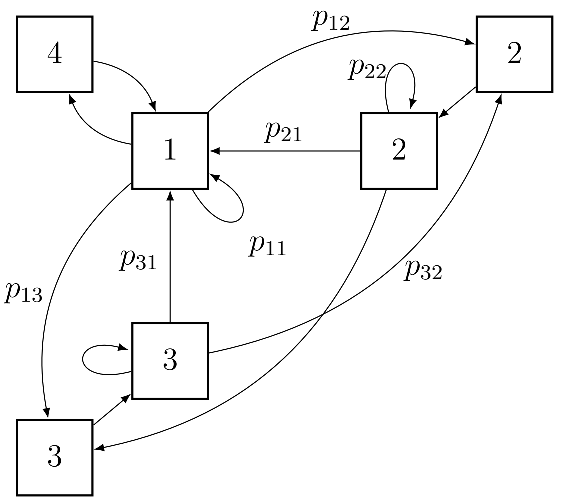

Figure 6: Transition diagram.

a good estimate as could the slope of known speed-ups or slow-downs. However, performers take liberties with these decisions, exactly the liberties we would like to discover. This suggests employing a switching model with a small number of discrete states.

We use a Markov model for S on four states with transition probability diagram given by Figure 6. The 4 switch states correspond to 4 different behaviors for the performer: (1) constant tempo, (2) speeding up, (3) slowing down, and (4) single note stress. As shown in the diagram, we only allow certain transitions for musical reasons and for estimability. The marked transition probabilities are sufficient to infer the remainder. The fourth state, stress, corresponds to *tenuto*, a common feature of musical performance. Such stresses may be marked with a line over the note in question, but are more often a feature of the performer taste, corresponding to a longer-than-written duration of a particular note. Such emphases occur for a variety of musical purposes—emphasis of the beat in running notes, the top of a phrase, a "landing point" where a phrase ends, etc.—but are always within the frame of constant tempo. Thus we allow stress to occur only after and before notes in state 1. Furthermore, we cannot allow state 2 or state 3 to return immediately to state 1, or else "stress" could happen through these pathways. We impose related constraints for a transition from state 2 to state 3 and vice versa. Essentially, transitions into these states must remain there before leaving. Thus, the entire transition diagram is fully determined. This process can be viewed equivalently as a second order Markov chain. We discuss some potential improvements at the end of Section 3.

Our data gives y_i as the observed tempo (in bpm) of the note (or chord) of the i^{th} note onset in Chopin's Mazurka Op. No. 3. The hidden continuous variable (X_i) is taken to be a two component vector with the first component being the prevailing tempo and the second the amount of acceleration. The amount, or existence, of acceleration is determined by the current and previous switch states. We use l_i to denote the musical duration of a particular note as given by the written score, so, throughout this piece, a quarter-note (♩) has $l_i = 1/3$, an eighth note (♪) has $l_i = 1/6$, etc. This is because each measure contains three quarter notes. In more complicated music with changing time signatures or instances where the notation doesn't necessarily correspond with the time signature, more care would be required. The observed tempo is already normalized to account for variable note durations. When the performer is in state 1 (or transits in and out of state 4), we take the prevailing tempo as constant with no acceleration: $X_{i+1} = X_i$.

Corresponding to these configurations, the parameter matrices are given in Table 1. So for any performance, we want to be able to estimate the following parameters: σ^2_{tempo}, σ^2_{acc}, σ^2_{stress}, σ^2_{ϵ}, the probabilities of the transition matrix (there are 7), and means μ_{tempo}, μ_{acc}, and μ_{stress}. Lastly, we have the initial state distribution

$$x_1 \sim N\left(\begin{pmatrix}\mu_1\\0\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & 0\\0 & 0\end{pmatrix}\right) \quad \text{where} \quad s_1 = 1.$$

2.4 Estimation and computational issues

To understand the performance decisions of individual musicians, we wish to simultaneously learn θ, S, and X. Because the switch states S and the continuous states X are both hidden, this becomes an NP-hard problem. In particular, there are 4^n possible paths through the switch variables, so evaluating the likelihood to maximize over θ via Algorithm 1 at each path is intractable. Ghahramani and Hinton (2000) give a variational approximation to estimate θ without also estimating S, but, as our goal is to learn both, we use the particle filtering approximation described in (Fearnhead and Clifford, 2003). Whiteley et al. (2010) refer to this algorithm as the Discrete Particle Filter, and it can be seen as an instance of the "Beam Search" optimization technique (Bisiani, 1992). The details are given in Algorithm 3 but the

Switch states		Transition equation parameter matrices		
s_i	s_{i-1}	d	T	$RQR^\top$
1	1	0	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$
2	1	$\begin{pmatrix}l_i\mu_{\text{acc}}\\ \mu_{\text{acc}}\end{pmatrix}$	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\sigma^2_{\text{acc}}\begin{pmatrix}l_i^2 & l_i\\ l_i & 1\end{pmatrix}$
3	1	$\begin{pmatrix}-l_i\mu_{\text{acc}}\\ -\mu_{\text{acc}}\end{pmatrix}$	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\sigma^2_{\text{acc}}\begin{pmatrix}l_i^2 & l_i\\ l_i & 1\end{pmatrix}$
4	1	$\begin{pmatrix}0\\ \mu_{\text{stress}}\end{pmatrix}$	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\begin{pmatrix}0 & 0\\ 0 & \sigma^2_{\text{stress}}\end{pmatrix}$
2	2	0	$\begin{pmatrix}1 & l_i\\ 0 & 1\end{pmatrix}$	$\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$
3	2	$\begin{pmatrix}-l_i\mu_{\text{acc}}\\ -\mu_{\text{acc}}\end{pmatrix}$	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\sigma^2_{\text{acc}}\begin{pmatrix}l_i^2 & l_i\\ l_i & 1\end{pmatrix}$
1	2	$\begin{pmatrix}\mu_{\text{tempo}}\\ 0\end{pmatrix}$	0	$\begin{pmatrix}\sigma^2_{\text{tempo}} & 0\\ 0 & 0\end{pmatrix}$
3	3	0	$\begin{pmatrix}1 & l_i\\ 0 & 1\end{pmatrix}$	$\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$
2	3	$\begin{pmatrix}l_i\mu_{\text{acc}}\\ \mu_{\text{acc}}\end{pmatrix}$	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\sigma^2_{\text{acc}}\begin{pmatrix}l_i^2 & l_i\\ l_i & 1\end{pmatrix}$
1	3	$\begin{pmatrix}\mu_{\text{tempo}}\\ 0\end{pmatrix}$	0	$\begin{pmatrix}\sigma^2_{\text{tempo}} & 0\\ 0 & 0\end{pmatrix}$
1	4	0	$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$	$\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$

Switch states		Measurement equation parameter matrices		
s_i		c	Z	G
4		0	$\begin{pmatrix}1 & 1\end{pmatrix}$	σ^2_ϵ
else		0	$\begin{pmatrix}1 & 0\end{pmatrix}$	σ^2_ϵ

Table 1: Parameter matrices of the switching state space model.

Algorithm 3 Discrete particle filter

```
Input: $Y$, $\theta$, $\pi_1$ probability vector over initial states (paths), $B$ maximum beam width
for $i = 1$ to $n$ do
    Set $b_i = |\{\pi_i > 0\}|$, the number of current paths
    Use Algorithm 1 to calculate the 1-step likelihood $\ell_i$ for each path and every potential state $s_{i+1}$ resulting in $b_i|S|$ particles
    Set $\pi_{i+1} \leftarrow \pi_i \ell_i p_i$: multiply the path probability by the likelihood and the probability of transitioning. Normalize $\pi$.
    Set $b_{i+1} = |\{\pi_{i+1} > 0\}|$ . If $b_{i+1} > B$, resample the weights to get $B$ non-zero weights and renormalize
end for
Return $B$ paths $\{S_b\}_{b=1}^{B}$ along with their weights $\pi_n$.
```

intuition is as follows: (1) for the first few time points, evaluate one step of the Kalman filter for each possible subsequent discrete state and store all these values; (2) calculate weights for each path by updating previous weights with the likelihood and the transition probability; (3) continue through time until the number of stored values exceeds some threshold storage limit; (4) from that point forward, subselect the "best" paths using a sampling scheme. These paths can be selected greedily, retaining only the highest values to that point, though we use the resampling procedure of (Fearnhead and Clifford, 2003) which is designed to approximate to the full discrete distribution over paths with a subset of support points by minimizing the mean squared error.

Algorithm 3 returns B paths along with their weights through the discrete state S for a particular parameter value θ. One can view this as a (approximate) distribution over paths conditional on θ. Instead, we will simply take the path with the highest weight for inference via penalized maximum likelihood. Thus, the likelihood of a particular parameter vector θ is evaluated by computing the best path with Algorithm 3 and then using the best path with Algorithm 1.

2.5 Penalized maximum likelihood

Even without the latent discrete states, parameter estimation in state-space models is a difficult problem, often plagued by spurious local minima and non-identifiability. The addition of discrete states only exacerbates this issue. However, for the present application, we have reasonably strong prior information for many of the parameters. The three mean parameters

Parameter		Distribution	Prior mean
σ^2_ϵ	$\sim$	Gamma(40, 10)	400 bpm^2
μ_{tempo}	$\sim$	Gamma($\overline{Y}^2/100$, $100/\overline{Y}$)	$\overline{Y}$ bpm
$-\mu_{\text{acc}}$	$\sim$	Gamma(15, 2/3)	10 bpm
$-\mu_{\text{stress}}$	$\sim$	Gamma(20, 2)	40 bpm
σ^2_{tempo}	$\sim$	Gamma(40, 10)	400 bpm^2
σ^2_{acc}	$=$	1	1 bpm^2
σ^2_{stress}	$=$	1	1 bpm^2
$p_{1\cdot}$	$\sim$	Dirichlet(85, 5, 2, 8)	
$p_{2\cdot}$	$\sim$	Dirichlet(4, 10, 1, 0)	
$p_{3\cdot}$	$\sim$	Dirichlet(5, 3, 7, 0)	

Table 2: Informative prior distributions for the music model

μ_{tempo}, μ_{acc} and μ_{stress} have sign restrictions in addition to strong information about their magnitude: average tempo should be around the indicated 132 bpm, the average amount of acceleration should probably be less than the size of a stress. We also have reasonably strong information about the probabilities of transitioning between states: self-transitions should be reasonably likely, long periods of speeding up are less likely than long periods of slowing down which are less likely than long periods in the constant tempo state. Because of this information, we use informative priors as penalties on all the parameters we estimate. This has the effect of introducing extra curvature to the optimization problem as well as conforming with musical intuition. The specific choices are shown in Table 2. We chose to fix σ^2_{acc} and σ^2_{stress} after numerical experiments suggested that they simply concentrated around their prior values independent of the other parameters.

3 Analysis of Chopin's Mazurka Op. 68 No. 3

We use the model and procedures developed above to estimate the parameters and performance choices for all 46 recordings of Chopin's Mazurka. Here we describe the inferences our model allows on some representative performances, describe parametric clusters determined by our model, contrast these with some alternative approaches to music modelling, and discuss some difficulties we encountered.

3.1 Musical analysis

Throughout his life, Frédéric Chopin, composed dozens of Mazurkas, of which 58 have been published. Inspired by a traditional Polish dance, these pieces gave Chopin an idiomatic style upon which to elaborate a wide variety of different compositional techniques, a practice German and Italian composers had employed frequently over the previous 3 centuries (cite Burkhardt). Repetition of themes, figures, or even small motives plays a central role in both the traditional dance and Chopin's compositions as do particular rhythmic gestures (cite Kallberg), especially the dotted-eighth sixteenth note pattern on the first beat of a measure.

Chopin's Op. 68 Mazurkas are a set of four similar works, published posthumously in 1855. The Op. 68 No. 3, which we analyze here, was composed in 1830, when Chopin was 20 years old. Around this time, Chopin, already a piano virtuoso and accomplished composer, left his native Warsaw and settled in Paris, where we would remain until his death in 1849.

This Mazurka has a rather simplistic ternary structure with two outer sections and a contrasting middle (ABA). The first A section is made up of four eight-bar phrases (*aaba*). The first phrase is echoed by the second phrase: they are nearly identical, with the two exceptions being that (1) the second is marked *piano* (soft) rather *forte* (strong) and (2) the second ends on the tonic (F major) rather than the dominant (C major). The fourth eight-bar phrase is an exact repetition of the second. The second A section is simply a repeat of the first two eight-bar phrases of the beginning. The intervening B section is 12 bars long, divided into three four-bar groups. The first four bars are simply a repeated interval of a perfect 5^{th} in the left hand. This ostinato will continue for the whole section. The remaining eight measures consist of a four-bar phrase repeated twice. The second time differs from the first only on the final two notes, preparing the recapitulation of the A section.

In terms of tempi, the B section is indicated to be faster, with the marking *Poco più vivo* (a little livelier). The B section ends with a ritardando into the following A section. The *b* section ends with a *fermatta* in measure 24, indicating an arbitrary elongation while the piece concludes with a two-measure long *ritardando*. Throughout, frequent markings prescribe emphasis of the third beat of each measure. This emphasis is also in keeping with the mazurka style, an intentional thwarting of the listener's expectation of first-beat emphasis.

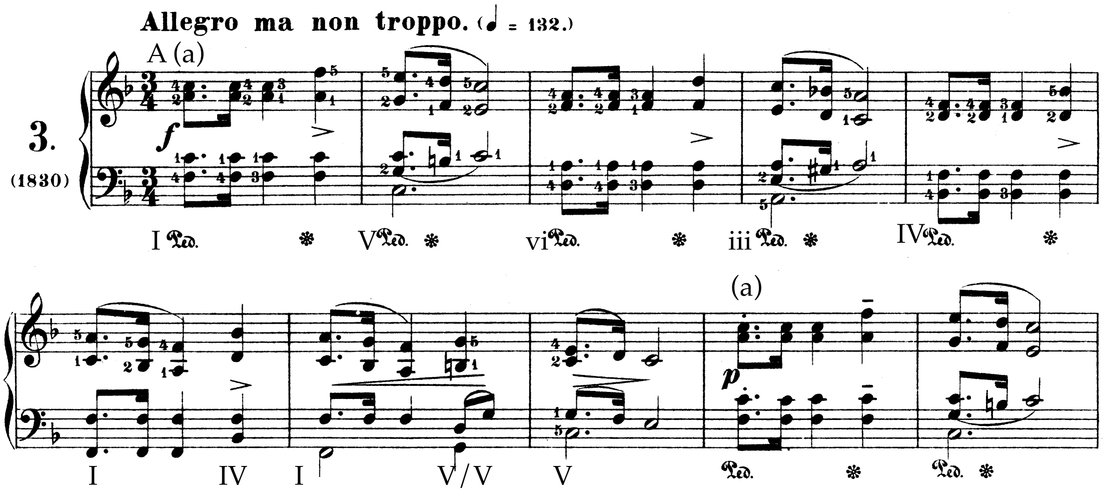

Figure 7: The first ten measures of Chopin's Mazurka Op. 68, No. 3. The harmonic progression is indicated below the staff in Roman numerals. Sections are marked above the staff, e.g. A (a). Analysis by the authors.

Figure 7 shows the first ten measures of the musical score with annotations for the sections discussed above and the harmonic progression in Roman numerals below the staff. The harmonies are standard, in fact, they are essentially the same as those of Pachelbel's *Canon*, familiar to many as "that song played at weddings." This harmonic progression, combined with the rhythmic repetition suggests a further division of this section, and all analogous sections, into three small groupings: two two-measure phrases, followed by a four measure phrase.

As a performer, these harmonic, rhythmic, and structural analyses aid in interpretation. The performer needs to decide how to emphasize or deemphasize these demarcations with slight or overt tempo or dynamic alterations. In a live performance, she could use physical motion to further suggest a particular interpretation. She can choose to emphasize long phrases, in this case, phrases of eight measures, or the shorter sub-phrases. Because of the repetition of similar phrases, she may choose to emphasize the long phrase on the first occurrence and shorter sub-phrases later on for variety. While the musical structure suggests such possible interpretations, the performer must make these choices on their own, and may even alter those decisions from performance to performance.

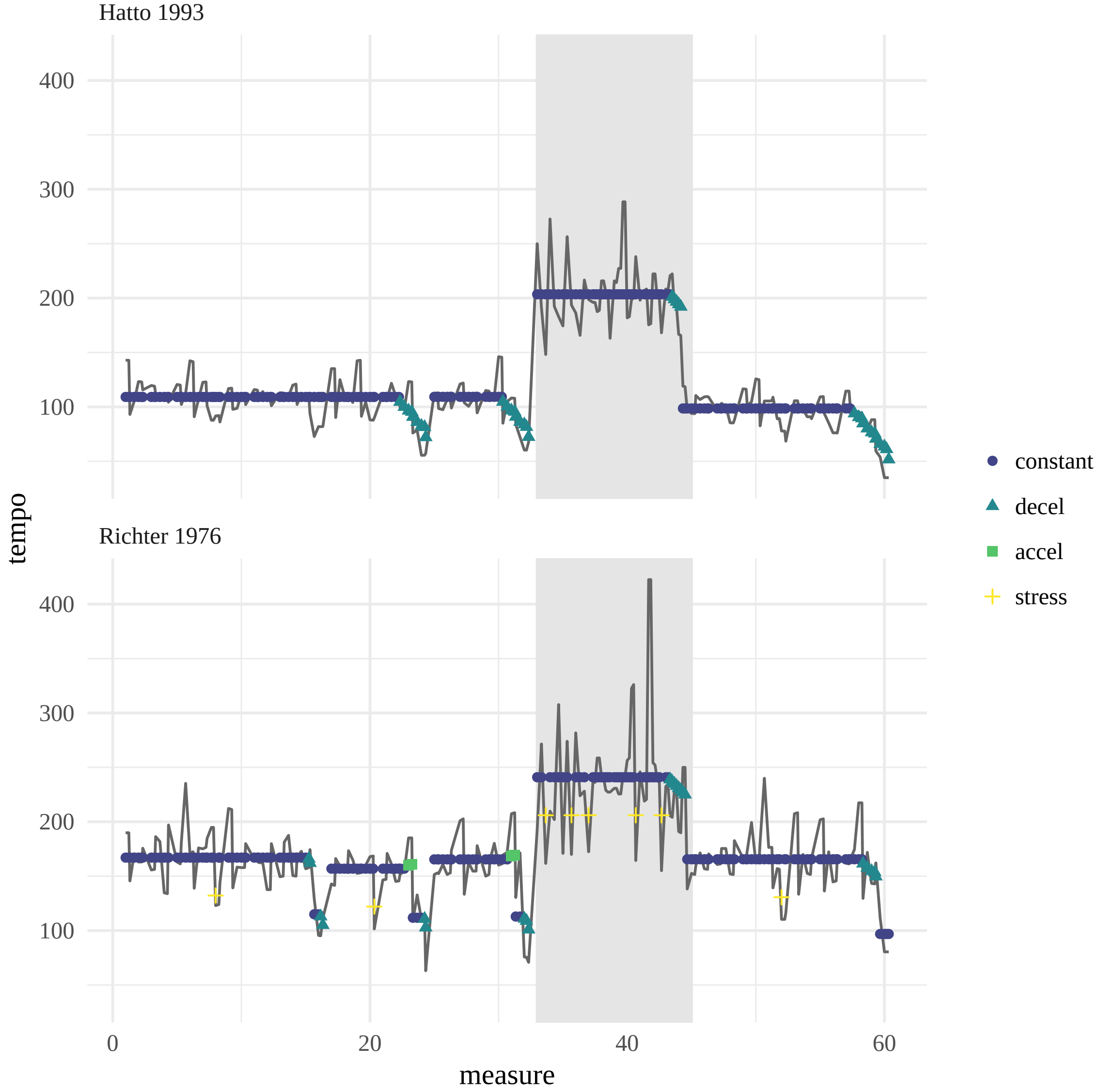

Figure 8: Inferred performer choices for two recordings.

3.2 Archetypal performances

Here we will carefully investigate how our model learns interpretive decisions for two rather different performances. Figure 8 shows the inferred state sequence for recordings made by Joyce Hatto in 1993 and Sviatislav Richter in 1976. The B section is shaded in gray to better illustrate the formal divisions discussed above.

In terms of our model, these two performers are quite different from each other. Hatto maintains a constant tempo carefully, remaining in state 1 with the exception of four periods of deceleration. All four periods coincide with the most significant phrase endings: at the

Table 3: The estimated parameters for performances by Richter and Hatto.

	σ^2_ϵ	μ_{tempo}	μ_{acc}	μ_{stress}	σ^2_{tempo}	p_{11}	p_{12}	p_{22}	p_{31}	p_{13}	p_{21}	p_{32}
Richter 1976	426.70	136.33	-11.84	-34.82	439.38	0.85	0.05	0.74	0.44	0.02	0.25	0.17
Hatto 1993	405.57	130.36	-13.57	-27.93	408.99	0.94	0.03	0.82	0.36	0.01	0.16	0.19

end of the A section at measure 32, the end of the B section at measure 48, at the end of the piece, and the minor transition from $b \to a$ in the first A section (measure 24). According to our inferred model, she never accelerates or uses the transitory stress state.

In contrast, Richter uses all four states from our model. The short blips of acceleration before the B section and before the $b \to a$ transition are slightly out of place, and are likely better labelled as "constant", but these state transitions describe more severe decelerations than the model's linear assumption would allow. Richter uses stress frequently. Some may well be attributable to larger variance around constant tempo (picked up as frequent stress rather than larger σ^2_ϵ), but most correspond to interesting note emphases, for example the second beat of measure 20. This note is essentially a minor phrase ending, but is also marked in the score with a *sforzando* (with sudden emphasis). It's the first of two such occurrences in the piece, the second coming four measures later on the *fermatta*, Richter's slowest note in the entire piece. Richter likely chooses to make this prescribed emphasis with a sudden slow down in part because it takes place within the context of an already loud passage, precluding the use of extra volume. Figure 3.2 shows the estimated parameters for these two performances. Richter has larger observation variance, σ^2_ϵ, slightly faster average tempo, lower acceleration, and larger stress. He also has a larger tempo variance, meaning that returns to state 1 can start at relatively different tempos. On the other hand, Hatto is much more likely to remain in states 1 or 2. These inferences are largely consistent with the visual takeaways of Figure 8. It's easy to see the increased variability around the "constant tempo" in Richter's performance and the higher overall tempos in both the A and B sections.

While these two performances are quite different from each other, they also display some similarities. Both take a faster tempo in the B section versus the A sections. Both performers slow down at the end of the piece, at the end of the B section, immediately preceding the B section, and at the $b \to a$ transition.

3.3 Clustering musical performances

To better understand how all the 46 performances relate to each other, we applied parametric clustering using the eleven-dimensional vector of estimated parameters. Because the estimated parameters lead to data distributions, we use the squared Hellinger distance to relate performances. For densities f and g which are absolutely continuous with respect to the dominating measure λ,

$$\begin{aligned} H^2(f,g) &= \frac{1}{2}\int\left(\sqrt{f(x)}-\sqrt{g(x)}\right)^2 d\lambda(x) = 1-\int\sqrt{f(x)g(x)}d\lambda(x) \\ &= 1-\int\sqrt{f_1(x_1)g_1(x_1)}d\lambda(x_1)\int\sqrt{f_2(x_2)g_2(x_2)}d\lambda(x_2), \end{aligned}$$

if $f(x) = f_1(x_1)f_2(x_2)$ and $g(x) = g_1(x_1)g_2(x_2)$.

4 Discussion

SUPPLEMENTARY MATERIAL

R-package "dpf": R-package containing code to perform the methods described in the article. The package also contains all data sets used as examples in the article. (GNU zipped tar)

References

Anderson, B. D. and Moore, J. B. (1979), *Optimal filtering*, Prentice-Hall, Englewood Cliffs, NJ.

Andrieu, C., Doucet, A. and Holenstein, R. (2010), 'Particle Markov chain Monte Carlo methods', *Journal of the Royal Statistical Society. Series B, Statistical Methodology* **72**(2), 1–33.

Bernstein, L. (2005), *Young People's Concerts*, Amadeus Press, Pompton Plains, NJ.

Bisiani, R. (1992), *Encyclopedia of Artificial Intelligence*, 2nd edn, John Wiley and Sons, chapter Beam Search.

Block, B. A., Jonsen, I. D., Jorgensen, S. J., Winship, A. J., Shaffer, S. A., Bograd, S. J., Hazen, E. L., Foley, D. G., Breed, G., Harrison, A.-L. et al. (2011), 'Tracking apex marine predator movements in a dynamic ocean', *Nature* **475**(7354), 86.

Dror, G., Koenigstein, N., Koren, Y. and Weimer, M. (2012), The Yahoo! music dataset and KDD-Cup'11, *in* 'KDD Cup', pp. 8–18.

Durbin, J. and Koopman, S. (2001), *Time Series Analysis by State Space Methods*, Oxford Univ Press, Oxford.

Durbin, J. and Koopman, S. J. (1997), 'Monte Carlo maximum likelihood extimation for non-Gaussian state space models', *Biometrika* **84**(3), 669–684.

Fearnhead, P. and Clifford, P. (2003), 'On-line inference for hidden markov models via particle filters', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65**(4), 887–899.

Forsén, S., Gray, H. B., Lindgren, L. O. and Gray, S. B. (2013), 'Was something wrong with beethoven's metronome?', *Notices of the AMS* **60**(9).

Fox, E. B., Sudderth, E. B., Jordan, M. I. and Willsky, A. S. (2011), 'A sticky HDP-HMM with application to speaker diarization', *The Annals of Applied Statistics* **5**(2A), 1020–1056.

Fuh, C.-D. (2006), 'Efficient likelihood estimation in state space models', *Annals of Statistics* **34**(4), 2026–2068.
URL: *http://arxiv.org/abs/math/0611376*

Ghahramani, Z. and Hinton, G. E. (2000), 'Variational learning for switching state-space models', *Neural computation* **12**(4), 831–864.

Hamilton, J. (2011), 'Calling recessions in real time', *International Journal of Forecasting* **27**, 1006–126.

Harvey, A. C. (1990), *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press.

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering* **82**(1), 35–45.

Kim, C.-J. (1994), 'Dynamic linear models with markov-switching', *Journal of Econometrics* **60**(1-2), 1–22.

Kim, C. and Nelson, C. (1998), 'Business cycle turning points, a new coincident index, and tests of duration dependence based on a dynamic factor model with regime switching', *Review of Economics and Statistics* **80**(2), 188–201.

Kitagawa, G. (1987), 'Non-Gaussian state-space modeling of nonstationary time series', *Journal of the American Statistical Association* pp. 1032–1041.

Kitagawa, G. (1996), 'Monte Carlo filter and smoother for non-Gaussian nonlinear state space models', *Journal of Computational and Graphical Statistics* pp. 1–25.

Koyama, S., Pérez-Bolde, L. C., Shalizi, C. R. and Kass, R. E. (2010), 'Approximate methods for state-space models', *Journal of the American Statistical Association* **105**(489), 170–180.

McDonald, D. J. and McBride, M. (2018), 'Discovering musical interpretations with markov-switching state space models', in preparation.

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O. and Matthiopoulos, J. (2008), 'State–space models of individual animal movement', *Trends in ecology & evolution* **23**(2), 87–94.

Rauch, H. E., Striebel, C. and Tung, F. (1965), 'Maximum likelihood estimates of linear dynamic systems', *AIAA journal* **3**(8), 1445–1450.

Whiteley, N., Andrieu, C. and Doucet, A. (2010), Efficient bayesian inference for switching state-space models using discrete particle markov chain monte carlo methods, Technical Report 10:04, Bristol University.
URL: *https://arxiv.org/abs/1011.2437*