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

September 4, 2018

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, 2016).

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 doesnt 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 soloists performance and failing to understand the players 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.

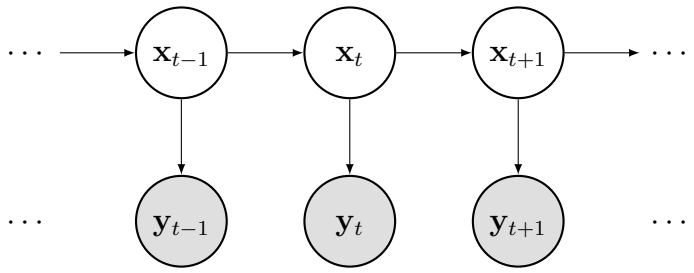

Figure 2: State-space model. Filled objects are observed, circles indicate that both hidden and observed states are continuous.

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:

$$y_t = f_\theta(x_t, \epsilon_t) \qquad x_{t+1} = g_\theta(x_t, \eta_t) \qquad x_1 \sim F, \tag{1}$$

where ϵ_t are η_t are marginally independent and identically distributed (IID) as well as mutually independent. Both y_t and x_t can be (generally) vector-valued, though in our application, y_t will be univariate. The vector $\{y_t\}_{t=1}^n$ is observed, and the goal is to make inferences for the unobserved states $\{x_t\}_{t=1}^T$ as well as any unknown parameters θ characterizing f_θ, g_θ, and the distributions of ϵ_t and η_t. 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_t and η_t normally distributed, (1) specializes

Algorithm 1 Kalman filter: estimate x_t conditional on $\{y_j\}_{j=1}^t$, for all $t = 1, \ldots, n$ 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 $t = 1$ to n **do**
  $H = RQR'$ ▷ Effective state variance
  $\chi_t \leftarrow d + Tx_{t-1|t-1}, \quad P_t \leftarrow H + TP_{t-1|t-1}T'$ ▷ Predict current state
  $\widetilde{y}_t \leftarrow c + Z\chi_t, \quad F_t \leftarrow G + ZP_tZ'$ ▷ Predict current observation
  $v_t \leftarrow y_t - \widetilde{y}_t \quad K_t \leftarrow P_tZ'F^{-1}$ ▷ Forecast error and Kalman gain
  $x_{t|t} \leftarrow \chi_t + K_tv_t, \quad P_{t|t} \leftarrow P_t - P_tZ'K_t$ ▷ Update
  $\ell(\theta) = \ell(\theta) - v_t'F^{-1}v_t - \log(|F_t|)$
end for
return $\widetilde{Y} = \{\widetilde{y}_t\}_{t=1}^n$, $\chi = \{\chi_t\}_{t=1}^n$, $\widetilde{X} = \{x_{t|t}\}_{t=1}^n$, $P = \{P_t\}_{t=1}^n$, $\widetilde{P} = \{P_{t|t}\}_{t=1}^n$, $\ell(\theta)$

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

Input: χ, $\widetilde{X}$, P, $\widetilde{P}$, T, c, Z.
$t = n$,
$\widehat{x}_n \leftarrow \widetilde{x}_n$,
while $t > 1$ **do**
  $\widehat{y}_t \leftarrow c + Z\widehat{x}_t$, ▷ Predict observation vector
  $e \leftarrow \widehat{x}_t - \chi_t, \quad V \leftarrow P_t^{-1}$,
  $t \leftarrow t - 1$, ▷ Increment
  $\widehat{x}_t = \widetilde{x}_t + \widetilde{P}_tTVe$
end while
return $\widehat{Y} = \{\widehat{y}_t\}_{t=1}^n, \widehat{X} = \{\widehat{x}_t\}_{t=1}^n$

to

$$\begin{aligned} x_t &= d + Tx_t + R\eta_t, \quad \eta_t \sim N(0,\ Q), \quad x_1 \sim N(x_0,\ P_0). \\ y_t &= c + Zx_t + \epsilon_t, \qquad \epsilon_t \sim N(0,\ G) \end{aligned} \tag{2}$$

where the matrices d, T, R, c, Z, and G are allowed to depend on θ and can potentially vary (deterministically) with t. 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_t and P_t represent the mean and variance of the conditional distribution of the unobserved component given only the observations $\{y_j\}_{j=1}^t$: $\chi_t = \mathbb{E}[x_t \mid y_1 \ldots, y_t]$ and $P_t = \mathbb{V}[x_t \mid y_1, \ldots, y_t]$. 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 $\widehat{x}_t = \mathbb{E}[x_t \mid y_1, \ldots, y_n]$ 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, $s_t = k$ 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 in 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_t, s_{t-1}, along with the parameter vector θ, to determine the specific dynamics at time t:

$$\begin{aligned} x_1 &\sim N(x_0,\ P_0), \\ x_{t+1} &= d(s_t, s_{t-1}) + T(s_t, s_{t-1})x_t + R(s_t, s_{t-1})\eta_t, \quad \eta_t \sim N(0, Q(s_t, s_{t-1})), \\ y_t &= c(s_t) + Z(s_t)x_t + \epsilon_t, \qquad\qquad\qquad\quad \epsilon_t \sim N(0, G(s_t)). \end{aligned} \tag{3}$$

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

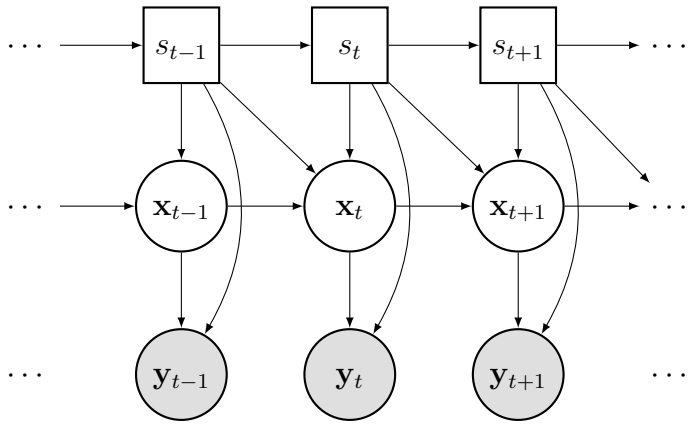

Figure 3: Switching state space model. Filled objects are observed, rectangles are discrete, and circles are continuous.

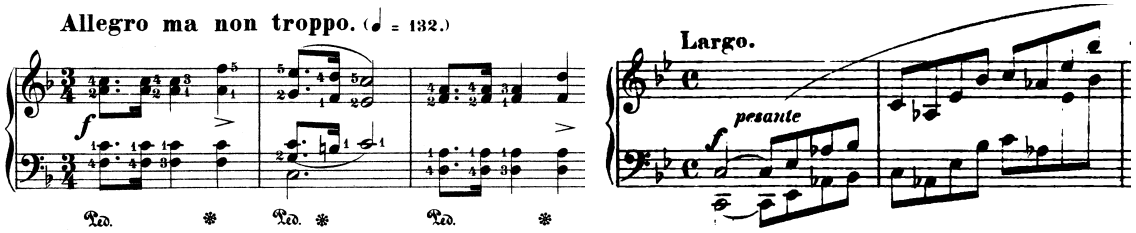

Figure 4: The beginning of two Chopin piano compositions: the Mazurka we analyze is on the left while the Ballade No. 1, Op. 23 is on the right.

govern the evolution of the system.

follows a hidden Markov model on four states, denoted $S_1, \ldots, S_4$ with transition probability diagram given by Figure 5.

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 simply marked *Largo*, which translates literally as "broad" or "wide".

Obviously, the metronome markings are much more exact, though even these are often viewed as approximations rather than commandments. 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.

We believe performers try to keep tempo steady for the majority of time. This is the most significant assumption we make, while this distinguishes our work from the previously mentioned approaches. However, a normal human being never plays precisely like a stop watch. The actual tempo the ratio between the actual time in seconds spent in a performance and the music time traversed in score is always changing. We think the actual tempi is the sum of the intended tempi a performer has in mind and the noises introduced by performance inaccuracy. For instance, the example in figure 5.3 is an excerpt with a single tempo marking, the solid lines represent a possible set of intended tempi, and the dotted lines are the actual tempi.

Estimating intended tempi does not seem to be very difficult at the first glance. If the boundaries of the tempo changes are known, then the average of tempi between boundaries would be a good estimate as could the slope derived though linear regression. But when the boundaries of the tempo changes are unknown, the problem becomes much more complicated than something solvable by regression. So we would like to use a switching model.

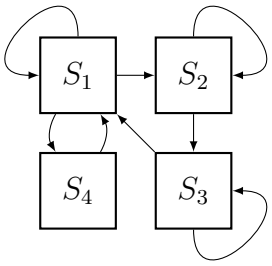

Figure 5: Transition diagram.

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. The hidden continuous variable (x_t) is taken to be a two component vector with the first component being the "ideal" tempo and the second being the acceleration. Corresponding to these configurations,

the parameter matrices are given in Table 1

Transition equation				
Switch states		Parameter matrices		
s_t	s_{t-1}	d	T	R
S_1	S_1	0	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
S_2	S_1	$\begin{pmatrix} l_t\tau_t \\ \tau_t \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & l_t \\ 0 & 1 \end{pmatrix}$
S_4	S_1	$\begin{pmatrix} 0 \\ \varphi_t \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
S_2	S_2	0	$\begin{pmatrix} 1 & l_t \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
S_3	S_2	$\begin{pmatrix} -l_t\tau_t \\ -\tau_t \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & l_t \\ 0 & 1 \end{pmatrix}$
S_1	S_3	$\begin{pmatrix} \mu_t \\ 0 \end{pmatrix}$	0	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
S_3	S_3	0	$\begin{pmatrix} 1 & l_t \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
S_1	S_4	0	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
Measurement equation				
Switch states		Parameter matrices		
s_t		c	Z	G
S_4		0	$\begin{pmatrix} 1 & 1 \end{pmatrix}$	σ_ϵ^2
else		0	$\begin{pmatrix} 1 & 0 \end{pmatrix}$	σ_ϵ^2

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

Finally,

$$Q = \begin{cases} \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_4^2 \end{pmatrix} & (s_t,\ s_{t-1}) = (S_4,\ S_1) \\ \begin{pmatrix} \sigma_3^2 & 0 \\ 0 & \sigma_4^2 \end{pmatrix} & (s_t,\ s_{t-1}) = (S_1,\ S_3) \\ \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} & \text{else.} \end{cases}$$

So for any performance, we want to be able to estimate the following parameters: σ_1^2, σ_2^2, σ_4^2, σ_ϵ^2, the probabilities of the transition matrix (there are 4), and vectors μ, τ, and φ. These last three will be of different lengths depending on the number of times the state is visited.

Lastly, we have the initial state distributions

$$x_1 \sim \begin{cases} N\left(\begin{pmatrix}\mu_1\\0\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & 0\\0 & 0\end{pmatrix}\right) & s_1 = S_1 \\ N\left(\begin{pmatrix}\mu_1\\\tau_1\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & 0\\0 & \sigma_2^2\end{pmatrix}\right) & s_1 = S_3. \end{cases}$$

Importantly, this is just one way to write this model.

2.4 Penalized likelihood maximization

2.5 Computational issues

The problem with estimating a model like this is that, because the switch states and the continuous states 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 at all of them is intractable. Thus, I implemented a particular approximation. The Beam Search (Algorithm 3), finds (greedily) evaluates the most likely path through the switch states. Another name for the algorithm is Discrete Particle Filter (`dpf`). Once we have those, `getLogLike` returns the negative loglikelihood of the data associated with that path. So for any configuration of parameters, we would form the matrices (`yupengMats`) then find the best path (`dpf`) then evaluate the likelihood of that path (`getLogLike`). We can then optimize over parameters using any variety of numerical optimization technique. However, when I have tried this, I always get infinite likelihood.

For this model, the dpf is more easily specified if we make the measurement equation depend only on the current state and not the previous state. For this reason, the code uses 16 states rather than 4. One can always change a Markov model in this way.

Algorithm 3 Beam search

1: **Input:** Initial parameters of the matrices. Integer beam width B.
2: **for** $i = 1$ **to** N **do**
3: (`dpf` performs 1-step of the following);
4: For each current path, calculate the 1-step likelihood for moving to each potential switch (`kf1step`)
5: Multiply the likelihood by the probability of transitioning to that switch state
6: Multiply by the previous path weights w
7: If $\|w\|_0 > B$, resample the weights (`resampleSubOptimal`) to get B non-zero weights which add to 1.
8: Keep only those paths corresponding to the non-zero weights
9: **end for**
10: Return B paths through the switch space along with their weights.

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

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.

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.

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. (2016), 'Clustering classical music performances', 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.