Markov-switching State Space Models for Uncovering Musical Interpretation

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 discovering what makes different performances more or less appealing. Using information on both the tempo and dynamics of 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;

1 Introduction

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

Over the last few years, statistical analysis of recorded music has become more and more important to academics and in 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 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.

2 Background

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

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

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 Indjic (Cook and Sapp, 2009).

This application is especially fascinating since it allows for visual, numerical, and aural exploration of the effects of tuning parameter selection on inference. Working with Prof. Chris Raphael on a Bösendorfer CEUS reproducing piano in the IU Jacobs School of Music, we can capture detailed measurements of key and pedal trajectories over time. The information is precise enough to reproduce an accurate replica by artificially "playing" the piano just as was done during the original performance. The piano can also create and respond to MIDI data. However, statistical inferences for these data are difficult. Current procedures are ill-suited for parameter estimation in a computationally efficient manner, as it amounts to Gaussian mixture learning with K^n components. Existing optimization algorithms (Ghahramani and Hinton, 2000; Raphael, 2002) thus only approximate the global solution.

3 The main idea

- We want to model tempo and dynamic decisions.
- We want a musician to understand what the parameters mean.

We will use a switching state-space model as shown in Figure 2. For now, we assume s is a hidden Markov model on four states, denoted $S_1, \ldots, S_4$ with transition probability diagram given by Figure 3.

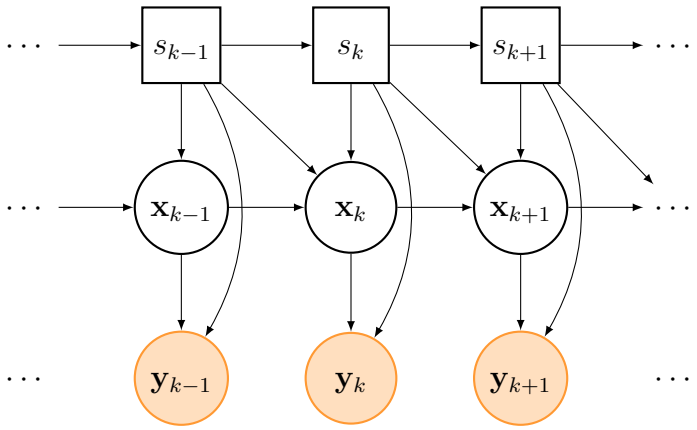

Figure 2: Switching state space model. Filled objects are observed, rectangles are discrete, and circles are continuous.

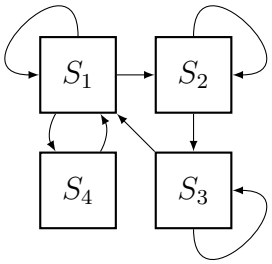

Figure 3: Transition diagram.

Models like this can have many behaviors, but for our case, the general form is:

$$x_t = d(s_t, s_{t-1}) + T(s_t, s_{t-1})x_t + R(s_t, s_{t-1})\eta_t \qquad \eta_t \sim N(0, Q(s_t, s_{t-1})) \tag{1}$$

$$y_t = c(s_t) + Z(s_t)x_t + \epsilon_t \qquad \epsilon_t \sim N(0, G(s_t)). \tag{2}$$

In other words, the hidden markov (switch) state determines which parameter matrices govern the evolution of the system.

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.

The `R` function `yupengMats` creates all of these different parameter matrices.

4 R documentation

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 1), 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.

SUPPLEMENTARY MATERIAL

Algorithm 1 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.

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

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

Cook, N. and Sapp, C. (2009), 'Purely coincidental? Joyce Hatto and Chopin's Mazurkas'.

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

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

McDonald, D. J. (2016), 'Clustering classical music performances', in preparation.

Raphael, C. (2002), 'A hybrid graphical model for rhythmic parsing', *Artificial Intelligence* **137**(1), 217–238.