Using Switching State-Space Models to Interpret Musical Dynamics

Robert Granger [*]
Department of Statistics, Indiana University

April 3, 2020

Abstract

The text of your abstract. 200 or fewer words.

Keywords: 3 to 6 keywords, that do not appear in the title

[*] The authors gratefully acknowledge . . .

1 Introduction

Things to write about in this section

1) Classifying music performances to sort on personal preference

2) Smoothing performances to sound better from Midi data

3) Using State-Switching Model

The composer gives directions about dynamics. p stands for "piano" and indicates this section should be played "soft". The "f" stands for "forte" and indicates this section should be played

2 The Switching State-Space Model

State-Space models are commonly used to model time series observations in the presence of perceived hidden, continuous states. As a result of the state-space framework, the observations are viewed as independent conditional on the hidden states whereas these hidden states will follow a vector autoregressive process. Adding assumptions of linearity and Gaussian error produces the following model commonly referred to as the general linear Gaussian state-space model Durbin & Koopman (2012) takes the form

$$\begin{aligned} y_t &= C_t + D_t x_t + \epsilon_t, \quad \epsilon_t \sim N(0, G_t) \\ x_{t+1} &= A_t + B_t x_t + \eta_t, \quad \eta_t \sim N(0, H_t), \quad x_1 \sim N(x_0, P_0) \end{aligned} \tag{1}$$

where the first part of Equation 1 is known as the observation equation and the second part is known as the state equation. y_t is a vector of known observations, and x_t is a vector of the unobserved, continuous states at each time period, t. The observation error, ϵ_t, and the state equation error, η_t, are both assumed to be independent and identically distributed.

In the typical state-space framework, the matrices A_t, B_t, C_t, D_t, G_t, and H_t are allowed to vary across time but are known. If there are a finite number of perceived structures for these matrices, and the given structure is unknown at time, t, a switching state-space model can be used. This model assumes there are some underlying discrete states, s_i, that transition over time through a Markov process. Making this slight adjustment to Equation 1 yields the following model which will be used as a basic framework for modeling music dynamics.

$$\begin{aligned} y_t &= C_t(s_t) + D_t(s_t)x_t + \epsilon_t, \quad \epsilon_t \sim N(0, G_t) \\ x_{t+1} &= A_t(s_t) + B_t(s_t)x_t + \eta_t, \quad \eta_t \sim N(0, H_t), \quad x_1 \sim N(x_0, P_0) \end{aligned} \tag{2}$$

When it comes to musical dynamics, rarely do musicians attempt to play with the same dynamics or loudness throughout the entirety of a piece. While there may be dramatic changes occasionally throughout the performance, most of the time we expect the musician to steadily change the loudness from note to note. Furthermore, the musician may quietly or loudly play individual notes for emphasis. In order to model this behavior, we propose the following four discrete states:

s^1: The musician selects a new value for loudness.

s^2: The musician continues the dynamics in a steady way.

s^3: The musician plays a single note more loudly.

s^4: The musician plays a single note more softly.

The observation, y_t is the univariate loudness of the note at each time period, t. In order to allow the dynamics to progress steadily, the continuous hidden states, x_t, follow a process that allows for piece-wise quadratic displays. At each time period, x_t, is an array of length three,

$$x_t = (x_t^0, x_t^1, x_t^2),$$

where x_t^0 is the loudness, x_t^1 is the first order difference, and x_t^2 is the second order difference. Since we want to maintain this smooth progression even when the musician plays a single note more loudly or softly, states s^3 and s^4 are implemented by adding a constant in the observation equation as opposed to changing the state equation. The model is similar to that proposed in Gu & Raphael (2012), but extended to include these additional states. Table 1 shows the parameter matrices for the four states.

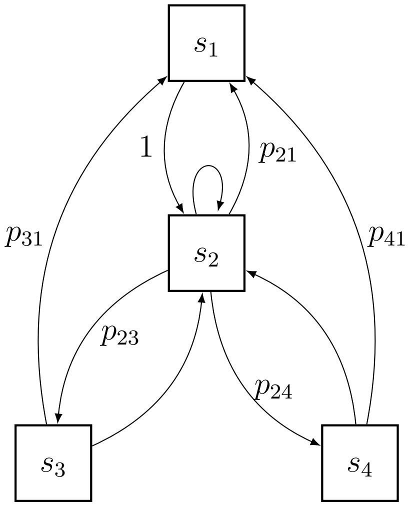

Figure 1: Transition diagram.

The final part of the switching state-space model is designing the Markov process for transitioning between the discrete states. Figure 1 displays the structure of the transition between states. The first state, s^1, allows for the selection of a new loudness which then transitions into the smooth progresson state, s^2, with probability 1. When arriving in s^2, the next time period's discrete state can be any of the possible states including itself. If in this progression, we move to a sudden loud note, s^3, or a sudden soft note, s^4, then we have the opportunity to continue the smooth progression, s^2, or start a new smooth progression, s^1. It is not permissible though to immediately then play another sudden loud or soft note. With four states, the transition matrix could potentially have 12 probabilities to estimate; however, because of restrictions placed on the possible transitions, only 5 probabilities need to be estimated.

In total, there are 13 distinct unknown parameters, $\theta \in \Theta$, spread across 4 unknown states, $s^i \in S$, through T time periods. A summary of the parameters to be estimated are: $\Theta = \{\mu_c, \sigma_\epsilon^2, \mu_0, \mu_1, \mu_2, \sigma_0^2, \sigma_1^2, \sigma_2^2, p_{21}, p_{23}, p_{24}, p_{31}, p_{41}\}$. In the next section, we discuss finding smoothed estimates of the dynamics which requires estimation of these parameters along with the continous and discrete states.

States	Parameter Matrices					
S	A	B	C	D	G	H
s^1	$\begin{pmatrix}\mu_0\\ \mu_1\\ \mu_2\end{pmatrix}$	$\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$	0	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	σ_ϵ^2	$\begin{pmatrix}\sigma_0^2&0&0\\0&\sigma_1^2&0\\0&0&\sigma_2^2\end{pmatrix}$
s^2	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}$	0	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	σ_ϵ^2	$\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$
s^3	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}$	μ_c	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	σ_ϵ^2	$\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$
s^4	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}$	$-\mu_c$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	σ_ϵ^2	$\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$

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

3 Evaluating the Model

3.1 The algorithm

The construction of the model is designed to allow us to separate the observed dynamics, y_t, from the performer's intended dynamics, $E[y_t]$. Hence, the aim in estimating the intended dynamics is attempting to remove ϵ_t, which can be seen as unintended deviations from the desired loudness.

When the parameter values, $\theta_i \in \Theta$, and the discrete states, S, are known, the Kalman filter (see Kalman (1960)) can be used to find estimates of the continuous hidden states, $\{x_t\}_{t=0}^{t}$, along with the likelihood of Θ. While the Kalman filter provides an easy way to compute the likelihood, the estimate of x_t is obtained using only observations coming before time t. In order to use all information, we implement the Kalman smoothing algorithm introduced in Rauch et al. (1965). Given Θ and S, this algorithm provides an estimate, $\hat{x_t} = E[x_t|y_0, ...y_T]$, which can be used to compute a fitted or "smoothed" value, $\hat{y_t}$.

So far, if Θ and S are known, we have a closed form solution to obtaining the smoothed dynamics through the Kalman smoother. If Θ is known, we can obtain the most likely set of discrete states, S, by simply running the Kalman smoother algorithm on every possible combination of discrete states and discovering the largest likelihood. This may work if the number of discrete states and/or time periods is small; however, the number of state combinations to check would be $\|\{s_i \in S\}\|^T$. For example, if we were using the music dynamics model presented in the previous section on Chopin's Mazurka Op. 63 No. 3 (which has 231 notes), the total number of state combinations to check would be $4^{231} \approx 1.19 \times 10^{139}$. Of course, many of these state combinations could be removed due to 0 likelihood given the restrictions on the state transitions, but even if this is taken into account, the number of state combinations would be far too large. To overcome this issue, we only look at a subset of these combinations through the Discrete Particle Filter as described in McDonald et al. (2019). The algorithm works by estimating the partial likelihood iteratively through time at each of the possible discrete state paths using the Kalman Filter. Each of these paths is known as a particle with only the previous states and partial likelihood being saved. This process would still require checking all $\{\#s_i \in S\}^T$ paths, so to get around this, a

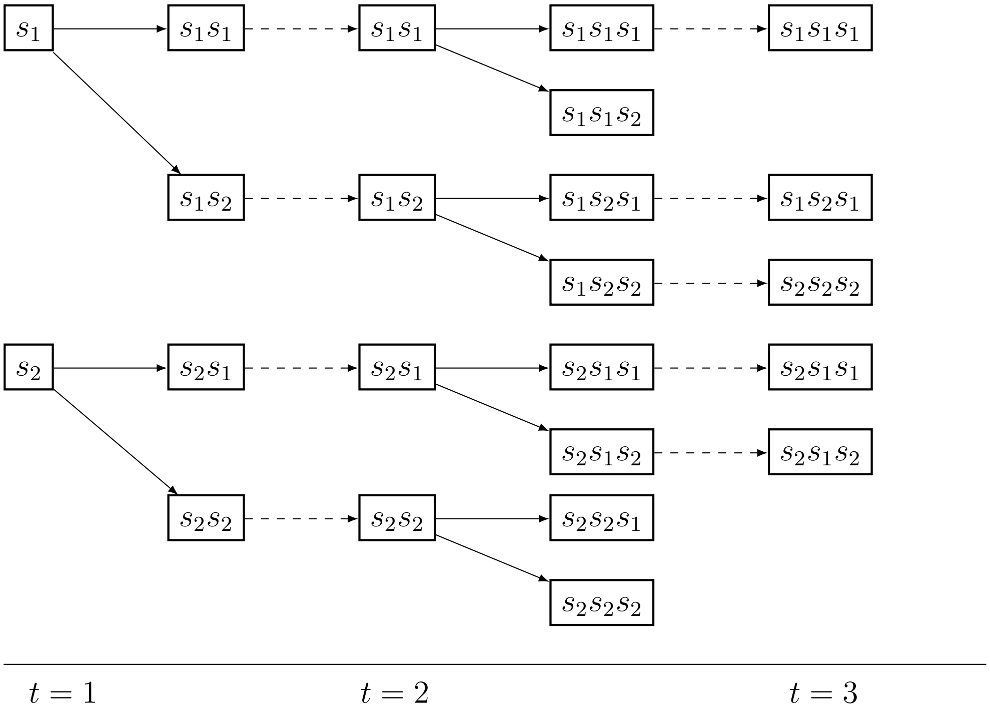

Figure 2: Discrete Particle Filter with 2 discrete states and capping the maximum number of stored particles at 5 for each time iteration.

maximum number of particles to save at each time iteration is selected.

Figure 2 illustrates this concept with 2 discrete states and the maximum number of particles set at 5. There are four possible state paths to check when passing from $t = 1$ to $t = 2$ which is less than the maximum particle number so all four paths are saved. When moving from $t = 2$ to $t = 3$, the number of paths increases to 8. Since only 5 paths are allowed, three of these paths must be dropped and will no longer be considered as possible solutions. In order to decide which paths are to be saved a sampling procedure must be performed. One sampling procedure proposed by Tugnait (1982) is to simply keep the paths that have the highest likelihood. Another sampling procedure proposed by Akashi & Kumamoto (1977) is to randomly sample one particle out of the total number of states descended from each of the $t - 1$ particles in proportion to their respective likelihoods. Fearnhead & Clifford (2003) proposes another stochastic approach but one that minimizes the expected mean squared error. This approach determines a threshold value such that all particles with likelihood above this value are kept and the remaining particles to be kept are chosen at random with probability equal to their respective likelihoods.

The final step is estimating the model parameters, Θ, that maximize the likelihood. This is no easy task as this model presents a couple of problems: 1) many of the parameters are constrained in our model as variances need to be positive and the probabilities should be between 0 and 1; 2) there are many local maxima making finding a global maxima difficult. To help alleviate these concerns, we can implement Bayesian priors on our parameters. Using carefully selected priors allows us to steer the algorithm away from impossible solutions and direct it toward more desired solutions. These issues also lead us to carefully consider our optimization technique. We will use two different methods. First, we will implement Nelder-Mead which can find local maxima for unconstrained multidimensional problems without the need for derivative computation Nelder & Mead (1965). Second, we will implement Simulated Annealing (SANN) which is commonly used when dealing with problems with many local maxima in order to estimate the global maxima. For both of these methods, the selection of initial values can greatly determine the outcome, so many should be selected and tried. The best solution found, regardless of the algorithm or initial values, shall be deemed the approximate solution.

The complete algorithm can be found in Algorithm 1. The solution to this model was solved using software R R Core Team (2019). The Nelder-Mead optimization is performed using the package `optimr` by Nash (2019) and the discrete particle filter was implemented by extending the package `dpf` by McDonald (2020).

```
Input Y, a vector of observed dynamics;
Initialize $\Theta$;
while (Stopping Criteria) do
    Create Matrices A, B, C, D, G, H at each $t$ for each $s^i \in S$;
    Implement Discrete Particle Filter;
    Compute the log-likelihood:  $l(Y|\Theta, S) + l(S|\Theta) + l(\Theta)$;
    Update $\Theta$ via Nelder-Mead Optimization or SANN;
end
```

Algorithm 1: Algorithm for Solving Music Dynamics Model

Parameter		Distribution	Prior Mean
μ_c	$\sim$	Gamma(100, 0.1)	10
σ_ϵ^2	$\sim$	Gamma(10, 0.5)	5
μ_0	$\sim$	Normal($\overline{Y}$, 10)	$\overline{Y}$
μ_1	$\sim$	Normal(0, 0.25)	0
μ_2	$\sim$	Normal(0, 0.25)	0
σ_0^2	$\sim$	Gamma(10, 1)	10
σ_1^2	$\sim$	Gamma(3, 1)	3
σ_2^2	$\sim$	Gamma(3, 1)	3
$p_{2,\cdot}$	$\sim$	Dirichlet(5, 85, 5, 5)	0.05, 0.85, 0.05, 0.05
$p_{3,\cdot}$	$\sim$	Beta(2, 8)	0.20
$p_{4,\cdot}$	$\sim$	Beta(2, 8)	0.20

Table 2: Informative prior distributions for Chopin's Mazurka Op. 63 No.3

3.2 Results for Chopin's Mazurka Op. 63 No. 3

The model was fit using observed dynamics from 46 different performances of Chopin's Mazurka Op. 63. No 3. This data was created by Andrew Earis Earis (2009) and found on the website for the Centre for the History and Analysis of Recorded Music (CHARM) CHARM (2009). The technique used to create the data is described in detail in his paper "An algorithm to extra expressive timing and dynamics from piano recordings" Earis (2007).

To implement Algorithm 1 on dynamics data from performances of Chopin's Mazurka Op. 63 No.3, we must make a few specific decisons. First, we choose prior distributions applicable specifically to Chopin's Mazurka Op. 63 No.3, which can be found in Table 2. Secondly, regarding the Discrete Partile Filter, we selected to keep up to 500 particles with the sampling criterion of keeping the largest likeliehoods. Lastily, since there are many local minima, we used 20 different starting points drawn randomly and independently from each of the prior distributions per optimization technique (both Nelder-Mead and SANN).

The results for one performance by Sviatislav Richter are shown in Figure 3 with the results of the other performances found in the appendix. The black line traces the observed

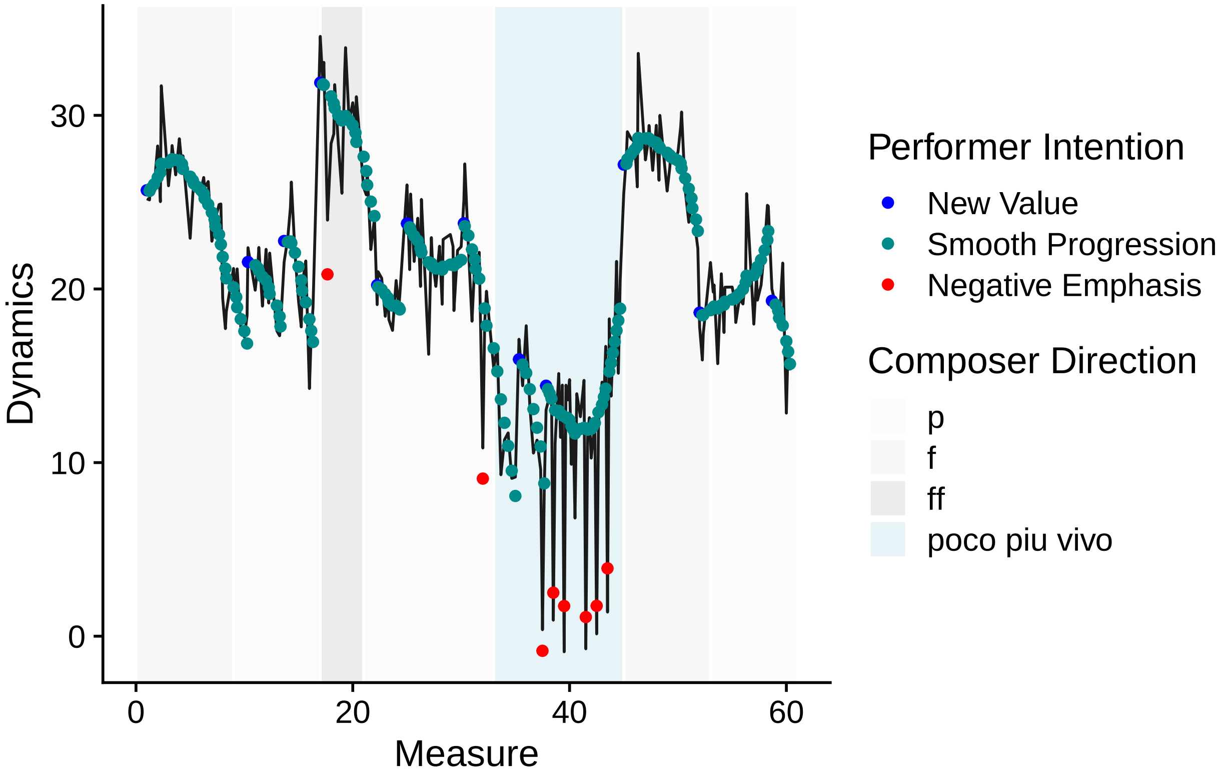

Figure 3: Interpreted Dynamics from Richter's 1976 performance of Chopin's Mazurka Op. 63 No. 3. The black line traces the observed dynamics while the colored dots are the smoothed interpreted dynamics.

dynamics for each note across time in accordance with their note onset. The colored dots are used to differentiate the 4 possible discrete states and show the interpreted musical dynamics for each note. The background is shaded based on musical directions given by the composer in the score. Notice with this performance, the model often seems to start a new smooth progression when the composer gives direction about the dynamics. There are occasions where this is not always the case. Notice that Richter appears to smoothly adjust the dynamics from the first "forte" section into the first "piano" section. Then he proceeds with a couple of abrupt changes to the dynamics although keeping it relavitely "soft". These unscripted parts of the performance are what we hope to discover as they can provide an explanation for why one person might prefer one performance over another.

μ_c	σ^2_ϵ	μ_0	μ_1	μ_2	σ^2_0	σ^2_1	σ^2_2
10.7833	4.2609	21.8404	-0.1353	-0.0222	12.2454	0.2225	0.0034
p_{21}	p_{23}	p_{24}	p_{31}	p_{41}			
0.0502	0.0165	0.0436	0.1068	0.04836			

Table 3: Parameter Estimates for Richter's 1976 Performance of Chopin's Mazurka Op. 63 No.3

The final important piece of this model that can be used to characterize a performance is the parameter values, Θ. The parameters for this particular performance can be found in Table 3, and all parameter estimates for the 46 performances can be found in the appendix. Certain parameters can be very informative in differentiating performances. For example, large values of μ_c will tell us that the performer often uses very loud or very soft points of emphasis, whereas smaller values indicate the performer is more subtle. If we want to get some idea as to the overall loudness of a piece, we may take a look at μ_0. The parameter values, along with the interpreted dynamics and discrete state, provide considerable information regarding differentiating between performances.

4 Conclusion

References

Akashi, H. & Kumamoto, H. (1977), 'Random sampling approach to state estimation in switching environments', *Automatica* **13**(4), 429–434.

CHARM (2009), *Centre for the History and Analysis of Recorded Music.*
URL: *http://www.charm.rhul.ac.uk/about/about.html*

Durbin, J. & Koopman, S. J. (2012), *Time Series Analysis by State Space Methods: Second Edition.*, Oxford University Press, Oxford.
URL: *http://grail.eblib.com.au/patron/FullRecord.aspx?p=991963*

Earis, A. (2007), ‘An Algorithm to Extract Expressive Timing and Dynamics from Piano Recordings’, *Musicae Scientiae* **11**(2), 155–182.
URL: *http://journals.sagepub.com/doi/10.1177/102986490701100202*

Earis, A. (2009), *Mazurka in F Major, Op. 68, No. 3.*
URL: *http://mazurka.org.uk/auto/earis/mazurka68-3/*

Fearnhead, P. & 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.
URL: *http://doi.wiley.com/10.1111/1467-9868.00421*

Gu, Y. & Raphael, C. (2012), ‘MODELING PIANO INTERPRETATION USING SWITCHING KALMAN FILTER’, *13th International Society for Music Information Retrieval Conference* .

Kalman, R. (1960), ‘A New Approach to Linear Filtering and Prediction Problems1’, *Journal of Basic Engineering* p. 12.

McDonald, D. J. (2020), *dpf: Discrete particle filtering.*
URL: *https://github.com/dajmcdon/dpf*

McDonald, D. J., McBride, M., Gu, Y. & Raphael, C. (2019), ‘Markov-switching State Space Models for Uncovering Musical Interpretation’, *arXiv:1907.06244 [stat]* .
URL: *http://arxiv.org/abs/1907.06244*

Nash, J. C. (2019), *optimr: A Replacement and Extension of the ’optim’ Function.*
URL: *https://CRAN.R-project.org/package=optimr*

Nelder, J. A. & Mead, R. (1965), ‘A Simplex Method for Function Minimization’, *The Computer Journal* **7**(4), 308–313.

R Core Team (2019), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

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

Tugnait, J. A. (1982), ‘Detection and estimation for abruptly changing systems’, *Automatica* **18**(5), 607–615.

Wickham, H. (2016), *ggplot2: Elegant graphics for data analysis*, 2nd edn, Springer.

Wickham, H. (2017), *tidyverse: Easily Install and Load the ’Tidyverse’*.
URL: *https://CRAN.R-project.org/package=tidyverse*