MODELING PIANO INTERPRETATION USING SWITCHING KALMAN FILTER

Yupeng Gu
Indiana University
School of Informatics and Computing

Christopher Raphael
Indiana University
School of Informatics and Computing

ABSTRACT

An approach of parsing piano music interpretation is presented. We focus mainly on quantifying expressive timing activities. A small number of different expressive timing behaviors (constant, slowing down, speeding up, accent) are defined in order to explain the tempo discretely. Given a MIDI performance of a piano music, we simultaneously estimate both discrete variables that corresponds to the behaviors and continuous variables that describe tempo. A graphical model is introduced to represent the evolution of the discrete behaviors and tempo progression. We demonstrate a computational method that acquires the approximate most likely configuration of the discrete behaviors and the hidden continuous variable tempo. This configuration represent a "smoothed" version of the performance which greatly reduces parametrization while retaining most of its musicality. Experiments are presented on several MIDI piano music performed on a digital piano. An user study is performed to evaluate our method.

1. INTRODUCTION

The score of Western classical music is a notation form that contains information such as pitches, durations, and words or symbols that give an abstract reference of how music should be played. It is rather a cartoon like description that misses much detail compared to actual performances. Classical musicians are trained to fill the differences. It is fair to say that the music we hear from CD or concerts have much more information than its corresponding notation. Often people use words such as intention, emotion, expression, interpretation, gesture, phrasing and articulation to describe this extra information. However, from a scientific point of view, these descriptions are vague, subjective and hard to quantify.

In this work, we propose a mathematical approach that aims to create a representation for interpretation. We think of interpretation as having a categorical component to it. This will be the discrete component of our model. We consider interpretation to be performance strategies for different groups of music notes, where the fine details of notes such as inter-onset intervals (IOI) in each group should be strongly correlated and explicitly constrained, instead of modeled independently. We believe this is similar to how musicians think of and communicate about music. Thus our representation will consist of discrete states that describe different performance behaviors and continuous variables that describe tempo and timing in detail. Where the detail will follow the characteristic of the discrete behaviors.

In order to circumvent the difficulty of audio recognition, we chose to use MIDI data for interpretation parsing in this work. The note by note detail and continuous controllers enable using MIDI to create and preserve expressive performances. For instance, we can find many MIDI piano performances from internet that demonstrate expressive interpretation. But relating MIDI data directly to interpretation is still not straightforward. Because the observable aspects of the performance are consequences of hidden interpretive notions. There is a missing layer of ideas that one needs in order to interpret the numbers in MIDI performance. This hidden layer of interpretive constructs guide the timing and volume data. We believe this hidden layer has a close relationship to interpretation and attempt to model it in this work.

This approach has many potential applications. In the area of creating expressive digital music in symbolic form, we have a long standing interest of trying to systematically change a performance meaningfully. It is often that someone had a decent performance recorded where some part of the performance is not fulfilling. If one is not willing or able to repeat the performance until getting better results, the only thing we can do is to modify parameters at the individual note level and hope some combinations might work. This is clearly an unnatural and unmusical way to modify an expressive performance. It would be better to operate on a higher-lever representation of the interpretation that understands notion of gestures and phrases. For instance, when we modify the parameter of a single note, some other parameters will compensate to retain a musical sense.

A musically meaningful representation of interpretation can also be used as a visualization tool. It is often an interesting experience for musicians to listen to a recording of themselves. As a listener, one has a different perspective and judges the performance more objectively. However, listening to a recording is time consuming, and we can only access a small amount of information at one time. Our representation can be used to visualize tempo changes in a discrete way, so musicians can take advantages of their eyes to see and explore an entire performance at once. Furthermore, such visualization can also be used to compare different performances, so it will be easier for musicians to discover how they differ from professionals.

Such representation could also be applied to the expressive rendering problem. With the development of the computer technology, there is a growing interest in generating performance that can match the level of professional musicians. The existing rendering systems are mostly rule-based or case-based. Such systems often include extracting and applying rules with parameters [1] [2] [3]. The advantage of our representation is that it is much lower dimensional than the usual MIDI performance. Hence it is easier to estimate the parameters rather than to estimate all the details for every note. Our representation also has the potential to reduce the unintentional activities from performers which could cause troubles in applying machine learning techniques to performances.

Another possible application of such representation is in creating accompaniment system. A traditional accompaniment system seeks to create a flexible accompaniment to a live soloist that follows the player [4] [5]. For most existing systems, the main focus is to keep up with the soloist as much as possible. Which could inevitably result in overfitting the soloists performance and failing to understand what the player's real intentions are. Good following requires a deeper understanding of the performers intention, thus separating signal from noise. Our representation has the potential to provide a performance model that maintains a certain level of musicality as well as offer enough flexibility. Also, a more advanced accompaniment system may be able to function like a music partner and even teach the player in the future. It is hard to imagine constructing such system without having a layer that can represent the interpretation.

We present a mathematical model in section 2. There is a literature on models that combine discrete state variables with Gaussian variables in fields such as economics, medical science and control engineering [6] [7] [8] [9]. These models are known alternately as Markov jump process, hybrid models, state-space models with switching and switching Kalman filter. We think this type of model suits our purpose of parsing the interpretation of a piano performance. A computation method is introduced in section 3 in order to compute the approximate most likely configuration of the variables in our model. Experiments are presented in section 4 as well as a brief user study that evaluates our model.

2. THE MODEL

We consider only expressive timing in this section. Suppose we have a music performance that contains a sequence of note onset times $o_0, o_1, ...o_N$. Let the score positions associated with the notes be $p_0, p_1, ...p_N$ which are measured in beats. We define four possible types of different behaviors regarding tempo activities. Every event will be labeled as one of the following four behaviors.

α_1. constant speed

In much notated music, especially Western classical music, often tempo marks or beats per measure(BPM) are used to indicate how fast the music should be played. It is clearly impossible for a human being to strictly execute them, but for most of the time these indications are still expected to be respected. There are words such as "rush" and "unstable" that sometimes are used by musicians to describe unintentional tempo change. In our analysis, we want to recognize and fix these unintentional actions.

α_2. slowing down

Although for many sections of music performance constant speed is intended, it is still very unlikely that such speed will be carried consistently though a music piece. An always strict in-tempo performance is often referred to as "mechanical", which is often undesirable and uncommon for Western classical music. Intentional tempo variation within a short time period is a technique that is often used to show expressiveness. Even though certain varying process could be very complicated, it can always be seen as a sequence of basic behaviors. We consider slowing down to be one of them.

α_3. speeding up

We consider speeding up to be the other basic behavior. Combined with α_1 and α_2, these three "devices" can theoretically represent any kind of tempo behaviors.

α_4. Accent (single note behavior)

A common technique to make an accent of a certain note is to take a little extra time before playing that note. Although it can also be seen as a slowing down followed by an immediate speeding up, in this discussion, we would like to model this as an individual behavior for two reasons: 1) Such behavior occurs often; 2) the tempi before and after accents are usually the same. We believe this needs to be modeled explicitly.

So, the possible discrete states for every event are described by the set $\Sigma = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. Our goal is to label each event o_n with a behavior S_n from Σ. Let $S_1, S_2, ..., S_N$ be the discrete behavior process, $S_n \in \Sigma, n = 1, ..., N$.

We model the sequence of the discrete states as a Markov chain. Figure 1 shows The Markov model. The assumptions are: 1) The states can stay in either constant speed state, slowing down state or speeding up state; 2) Before speeding up, there must be a slowing down process; 3) before slowing down, the performance must be in constant speed; 4) Accent can only happen during constant speed mode and will only last for one note. These assumptions are not necessarily true, we only make our model this way for simplicity.

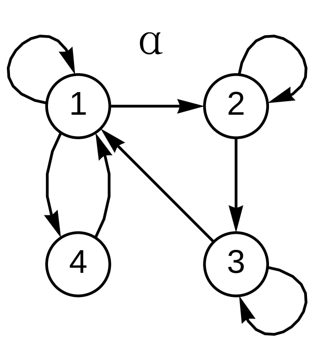

Figure 1. A Markov model showing possible transitions between the discrete states.

This Markov chain is modeled with initial probabilities:

$$I(s_1) = P(S_1 = s_1)$$

s_1 could only be α_1 or α_3 in our model – meaning we only start a performance with a constant tempo or speeding up to a constant tempo.

The transition probability matrix is defined as:

$$R(s_{n+1}, s_n) = P(S_{n+1} = s_{n+1} | S_n = s_n)$$

Now we model the tempo behavior in different states under a switching Kalman filter framework. Let $t_1, t_2, ..., t_N$ and $a_1, a_2, ..., a_N$ be the continuous variables that represent the tempo and acceleration associated with $o_1, o_2, ..., o_N$ repectively, measured in seconds per quarter note. Denote

$$X_{n,t} = t_n$$
$$X_{n,a} = a_n$$
$$X_n = (t_n, a_n)^T$$
$$l_n = p_n - p_{n-1}$$

Where l_n is the IOI in beats for two consecutive events. We have the initial distribution

$$X_1 \sim N(\begin{pmatrix} \mu_t \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix}) \qquad |_{S_1=\alpha_1}$$

$$X_1 \sim N(\begin{pmatrix} \mu_t \\ \mu_a \end{pmatrix}, \begin{pmatrix} \sigma_t^2 & 0 \\ 0 & \sigma_a^2 \end{pmatrix}) \qquad |_{S_1=\alpha_3}$$

Then we define the different behaviors

$$X_n = X_{n-1} \qquad |_{S_n=\alpha_1, S_{n-1}=\alpha_1} \quad (1)$$

$$X_n \sim N(\begin{pmatrix} \mu_t \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix}) \qquad |_{S_n=\alpha_1, S_{n-1}=\alpha_3} \quad (2)$$

$$X_n = X_{n-1} \qquad |_{S_n=\alpha_1, S_{n-1}=\alpha_4} \quad (3)$$

$$\begin{matrix} X_{n,a} \sim & N(\mu_a, \sigma_a^2) \\ X_{n,t} = & X_{n-1,t} + l_n X_{n,a} \end{matrix} \qquad |_{S_n=\alpha_2, S_{n-1}=\alpha_1} \quad (4)$$

$$X_n = \begin{pmatrix} 1 & l_n \\ 0 & 1 \end{pmatrix} X_{n-1} \qquad |_{S_n=\alpha_2, S_{n-1}=\alpha_2} \quad (5)$$

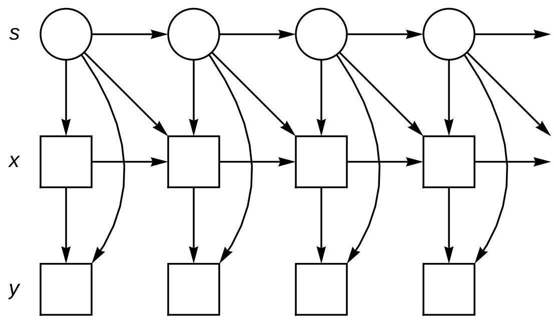

Figure 2. The DAG describing the dependency structure of the variables of our model. Circles represent discrete variables while squares represent continuous variables.

$$\begin{matrix} X_{n,a} \sim & N(-\mu_a, \sigma_a^2) \\ X_{n,t} = & X_{n-1,t} + l_n X_{n,a} \end{matrix} \qquad |_{S_n=\alpha_3, S_{n-1}=\alpha_2} \quad (6)$$

$$X_n = \begin{pmatrix} 1 & l_n \\ 0 & 1 \end{pmatrix} X_{n-1} \qquad |_{S_n=\alpha_3, S_{n-1}=\alpha_3} \quad (7)$$

$$X_n = X_{n-1} \qquad |_{S_n=\alpha_4} \quad (8)$$

This model forces the tempo to be a constant (but unknown) in each section where the discrete states stay in α_1 or α_4. It also forces an unknown constant acceleration when discrete states in α_2 and α_3. Equation (2) means when the performance comes back from speeding up, the performer will start a new unknown tempo. We denote all the unknown tempi $\{X_{n,t}$ s.t. $S_n = \alpha_1, S_{n-1} = \alpha_3\}$ as $\{\tau_1, ..., \tau_K\}$. Equation (4) means every time when the performance gets into slowing down states, a new unknown acceleration with a positive mean is introduced. Equation (6) means every time when the performance gets into speeding up states, a new unknown acceleration with a negative mean is introduced. We denote all the unknown accelerations $\{X_{n,a}$ s.t. $S_n = \alpha_2, S_{n-1} = \alpha_1$ OR $S_n = \alpha_3, S_{n-1} = \alpha_2\}$ as $\{\gamma_1, ..., \gamma_L\}$.

Now we relate this tempo and acceleration to the observables. Let the IOI $y_n = o_n - o_{n-1}$ for $n = 1, 2, ..., N$ Then the data model is:

$$y_n = l_n X_{n,t} + c_n + \epsilon_n \quad (9)$$

where

$$c_n = 0 \qquad |_{S_n \neq \alpha_4} \quad (10)$$

$$c_n \sim N(\mu_c, \sigma_c^2) \qquad |_{S_n=\alpha_4} \quad (11)$$

$$\epsilon_n \sim N(\mu_\epsilon, \sigma_\epsilon^2) \quad (12)$$

Equation (11) means when the performance comes to an accent, the performer will stretch the IOI with a random length. We denote all the unknown variables $\{C_n$ $s.t. S_n = \alpha_4\}$ as $\{\kappa_1, ..., \kappa_M\}$. Equation (12) represent the observation errors. All the other variables of the model depend deterministically on these variables

$$\tau_1, ..., \tau_K, \gamma_1, ..., \gamma_L, \kappa_1, ..., \kappa_M, \epsilon_1, ..., \epsilon_N$$

The directed acyclic graph(DAG) of the graphical model is represented in figure 2. The model has both discrete and continuous variables. For every configuration of the discrete variables, the continuous variable have a multivariate Gaussian distribution and is a Kalman filter. Thus, the $S_1, .., S_N, X_1, ..., X_N,\ y_1, ..., y_N$ collectively have a conditional Gaussian distribution.

3. COMPUTING THE INTERPRETATION PARSE

We want to simultaneously estimate the discrete state variable $S_1, S_2, ..., S_N$ and the continuous variable tempo $X_1, X_2, ..., X_N$ given the observed IOI data $y_1, y_2, ..., y_n$.

The joint likelihood function can be expressed as

$$L(y, s, x) = I(s_1)P(y_1|x_1, s_1) \times \prod_{n=2}^{N} (P(s_n|s_{n-1})P(x_n|x_{n-1}, s_n, s_{n-1})P(y_n|x_n, s_n))$$

We are interested in finding the best configuration of hidden variable S and X that has the greatest probability of giving the observation y.

$$(\hat{s}, \hat{x}) = \arg \max_{s \in S, x \in X} L(y, s, x)$$

Since our model has a linear graph structure described in Figure 2, the maximization problem can be solved using dynamic programming. Using the notation $a_i^j = \{a_i, a_{i+1}, ..., a_j\}$. Let $L_n(y_1^n, s_1^n, x_1^n)$ be the joint likelihood function for variables until observation n, y_1^n, S_1^n, X_1^n for $n = 1, 2, ..., N$. Then we define the density of the optimal configuration for variables until observation n

$$H_n(s_n, x_n) = \max_{s_1^{n-1}, x_1^{n-1}} L_n(y_1^n, s_1^n, x_1^n) \quad (13)$$

Then $H_n(s_n, x_n)$ can be computed recursively

$$H_1(s_1, x_1) = \max_{s_1} I(s_1)P(y_1|x_1, s_1)$$

$$\begin{aligned} H_n(s_n, x_n) = &\max_{s_{n-1}, x_{n-1}} H_{n-1}(s_{n-1}, x_{n-1}) \\ &\times P(s_n|s_{n-1}) \\ &\times P(x_n|x_{n-1}, s_n, s_{n-1}) \\ &\times P(y_n|x_n, s_n)), n = 2, ..., N \end{aligned}$$

We can see that

$$\max_{s_N, x_N} H_N(s_N, x_N) = \max_{s_1^N, x_1^N} L_n(y_1^N, s_1^N, x_1^N))$$

A more detailed description of this method can be found in [10].

It is obvious that the possible state sequences grow exponentially with the number of event N. In order to make the computation tractable, we need to approximate. In this experiment, we use a simple approach that is to sort the current hypotheses on probability densities and leave out the small ones in (13). This method is also known as "beam search".

Once we compute the approximately optimal configuration of discrete states $\hat{s}$. We can recover $\hat{x}$ from the Kalman filter defined by $\hat{s}$.

4. EXPERIMENTS

MIDI is our data format. We use the time stamps directly from MIDI files as the onset times of the notes. All data are collected from a high quality digital piano made by YAMAHA. The reason we choose such an instrument is to ensure that we can hear exactly the same thing as originally recorded when the music is being reproduced. Also when we evaluate our model by modifying the performances and compare them to the original ones, using this instrument can minimize the effect introduced by difference in sound characteristic. The piano keyboard is weighed to simulate the feeling of the real piano keys. According to the 5 pianists who helped creating the data, although it is still not the same as playing a real piano, they can adapt to it and play expressively.

3 sets of experiments are performed:

4.1 Smoothing a Performance

The first set of experiment demonstrates that our model parsimoniously and faithfully represents the original performance.

The data set contains 12 piano excerpts played by graduate level piano major students from the Indiana University Jacobs School of Music. In order to show the generality of our model, the excerpts are selected from composers from different time periods, including Bach, Haydn, Mozart, Beethoven, Schumann, Brahms and Barber. The notated tempi for the excerpts also differ (i.e. there are fast pieces and slow pieces.).

For each excerpt, we have a corresponding MIDI score created from music notation software. Using the method described in [11], we can acquire the music times $\{p_k\}$ in beats for the performance. For each note, we use the time stamp of the MIDI onset as our observation o_n. If several notes are struck at the same time (i.e. a chord), we use the onset time of the first note. μ_t is always set to be the notated tempo. $\sigma_t, \mu_a, \sigma_a, \mu_c, \sigma_c, \epsilon_n$ are manually set to some appropriate value.

Using the method described in section 2 and 3, we can compute the approximate optimal state configuration $\hat{s}$ and corresponding tempo process $\hat{x}_{\bullet,t}$. By reconstructing t, and hence y, from our estimated variables $\hat{x}_{n,t}$, we created a simplified or "smoothed" interpretation. Figure 3 4 5 shows some examples of observed IOIs and "smoothed" IOIs. In each figure, the top plot represents the tempo of the original performance where the bottom plot represents the "smoothed" version. There are many sections of the "smoothed" version that are horizontal lines, which is the behavior of α_1 constant tempo. The "peaks" in the bottom plot show the α_2 slowing down and α_3 speeding up expressive gestures as well as the α_4 accents. We want to see that if the "smoothed" version is approximating the original one with greatly reduced parametrization as well as capturing some of the "important events" and eliminates "unintentional variation".

We use the "smoothed" version to render a MIDI performance with everything else unchanged. Which includes MIDI velocities/note length for each notes and pedaling.

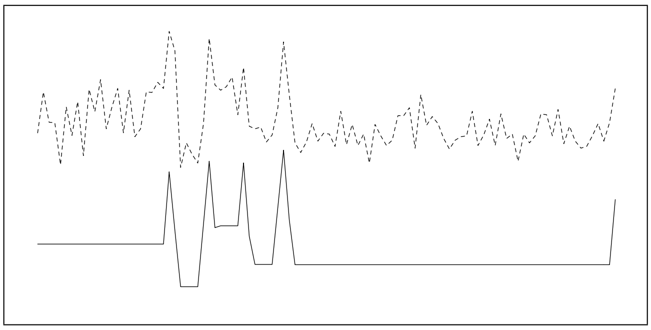

Figure 3. The original tempi and "smoothed" tempi of a performance of Schubert Piano Sonata D959, 1st movement excerpt. (The x-axis represent the music time as in beats. The dotted line represents the original performance. The normal line represents the "smoothed" version. Lines with slope $= 0$ represent state α_1; lines with slope > 0 represent state α_2; lines with slope < 0 represent state α_3; lines with slope $= +\infty$ represent state α_4)

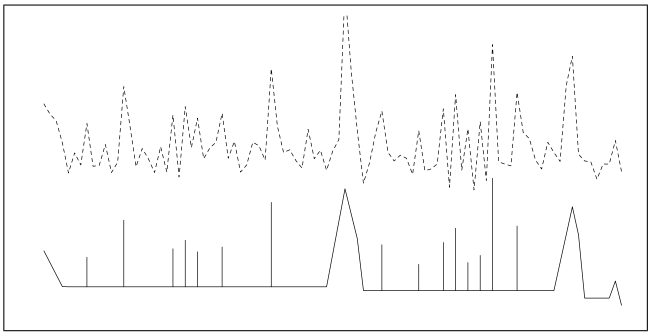

Figure 4. The original tempi and "smoothed" tempi of a performance of Beethoven Piano Sonata Op.31 No.3, 1st movement excerpt. (The lines have the same meaning as in Figure 3)

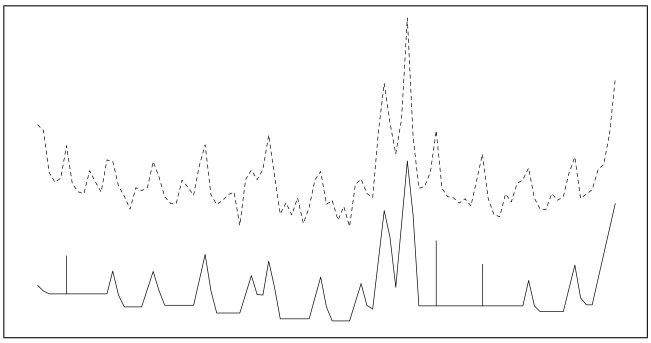

Figure 5. The original tempi and "smoothed" tempi of a performance of Chopin Etude Op.10 No.3 excerpt. (The lines have the same meaning as in Figure 3)

played by	worse	similar	better
professional pianists (4.1)	23	54	31
other musicians (4.2)	4	6	17
dynamic experiment (4.3)	1	7	1

Table 1. Results from the survey of asking 9 subjects about their opinions on "smoothed" version. The total 16 excerpts of the 3 sets of experiments are presented in random order to avoid bias.

Then we perform a simple user study. We present both the original version and the "smoothed" version to 9 participants who were graduate level piano major students. The subjects were presented with random ordering of the two versions of every excerpt. They are asked to choose one from the following options: 1) version 1 is better; 2) version 2 is better; 3) they are about the same. Although what we are really interested in is whether the "smoothed" version is similar to the original performance, we design the questionnaire this way to avoid putting bias towards choosing similar in our subjects' mind.

From the $9 \times 12 = 108$ results that evaluated in this way. The results are shown in table 1 Which shows many of the cases subjects think our "smoothed" version is at least on par with the original version.

4.2 Improving a Performance

The second set of experiment demonstrates that our model can provide a performance standard. If someone has a sense of musicality but lacks piano skills, our model may be able to improve their performance. This experiment differ from the previous one because the amateur piano players play less professionally. So we are testing the "correcting" and "improving" abilities of our model rather than "smoothing". This data set contains several excerpts played by students majoring string performance who knew music well but didn't have serious training in piano performance. We run the exact same procedure as in the first set of experiment and ask participants the same questions. From the results in table 1 we can see subjects think the "smoothed" version is better more often than the excerpts in 4.1, though we do not make inference on the larger population.

4.3 On Dynamics

The third set of experiment demonstrates that dynamics can also be modeled with the conditional Gaussian framework.

We choose Beethoven sonata op.101 1st movement as our material. First we manually partition our music into 3 monophonic voices. For each voice we have a series of MIDI velocities $v_1, v_2, ..., v_n$. Our model for dynamic has two types of behaviors β_1, β_2, in which dynamic can change to a new value or starting a new second order smooth progression.

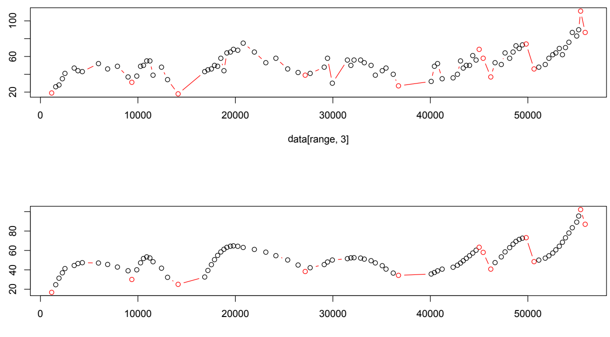

Figure 6. The original dynamics and "smoothed" dynamics of one voice of a performance of Beethoven Sonata Op.101, 1st movement excerpt.(Red dots represent state β_1; Black dots represent state β_2)

Denote

$$Z_n = (d_n, e_n, f_n)^T$$

We define the dynamic behaviors

$$Z_n = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} Z_{n-1}|_{R_{n-1}=\beta_2}$$

$$Z_n \sim N(\begin{pmatrix} \mu_d \\ \mu_e \\ \mu_f \end{pmatrix} \begin{pmatrix} \sigma_d^2 & 0 & 0 \\ 0 & \sigma_e^2 & 0 \\ 0 & 0 & \sigma_f^2 \end{pmatrix})|_{R_n=\beta_2, R_{n-1}=\beta_1}$$

$$Z_n \sim N(\begin{pmatrix} \mu_d \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \sigma_d^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix})|_{R_n=R_{n-1}=\beta_1}$$

Then we relate the model to observations

$$v_n = Z_{n,d} + \delta_n$$
$$\delta_n \sim N(\mu_\delta, \sigma_\delta^2)$$

Using the similar method described in section 3, we compute the "smoothed" dynamics. Figure 6 shows an example of the dynamics before and after the "smoothing".

We use both "smoothed" onset times and dynamics to render a new MIDI performance with everything else unchanged. Then we ask the subjects the same questions. Again, the majority think the "smoothed" version is at least on par with the original performance.

5. DISCUSSION

Although there is no clear evidence showing "smoothed" performances computed from our model are better than original ones, it is still interesting to see that people think they are comparable. It suggests that our model understands the interpretation in a reasonable way.

The direct follow-up of this work is applying the model in accompaniment system. It is challenging to deal with soloist's unintentional activities [11]. We can see from the experiments that our model can reduce the "performance noise". Also, learning the discrete parameters from our model may help with the score following problem since we can model following strategies separately in different parts of music.

In visualization scenario, our current model is only a "toy version" since it only has a limited number of behaviors. For future work, we will explore more "devices" (i.e. more discrete states) that match musicians' intuitive ideas.

Eventually, we also want to use such models for expressive rendering problem. For fine detail of performance, the "devices" may need to be more sophisticated than simple linear models.

We look forward to see more generally useful applications of this model framework as it develops.

6. REFERENCES

[1] Hiraga, R.; Bresin, R.; Hirata, K. & Katayose, H.: Rencon 2004: "Turing Test for Musical Expression" *Proceedings of International Conference on New Interfaces for Musical Expression, in Proc. of NIME 2004*, pp. 120-123, 2004

[2] G. Widmer, S. Flossmann, and M. Grachten, "YQX plays chopin", *AI Magazine*, vol. 30, no. 3, pp. 35-48, 2009.

[3] T. Suzuki, The Second Phase Development of Case Based Performance Rendering System Kagurame, *In Proc. of the IJCAI-03 Rencon Workshop*, 2003, pp. 17-25.

[4] C. Raphael, "Music Plus One and Machine Learning Machine Learning", *Proceedings of the Twenty-Seventh International Conference ICML* 2010.

[5] R. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Computer Music Conference, 1984*. Int. Computer Music Assoc., 1984, pp. 193 - 198.

[6] J.D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle". *Econometrica*, 57:357-384, 1989

[7] R.H. Shumway, D.S. Stoffer, "Dynamic linear models with switching". *J. Amer. Stat. Assoc.*, 86:763-769, 1991

[8] C.J. Kim, "Dynamic linear models with Markov-switching." *J.Econometrics*, 60:1-22, 1994

[9] Z. Ghahramani and G. Hinton. "Variational learning for switching state-space models". *Neural Computation*, 12(4):963-996, 1998.

[10] C. Raphael, "A Mixed Graphical Model for Rhythmic Parsing", *Proceed. of 17th Conf. on Uncertainty in Artif.* Int.462–471, Morgan Kaufmann, 2001.

[11] Y. Gu, C. Raphael, "Orchestral Accompaniment for a Reproducing Piano", *Proceedings of the ICMC09*, Montreal, Canada 501-504, 2009.