Online Music-to-Score Alignment with Hidden Semi-Markov Models

Gabriel Huang

January 14, 2016

Abstract

Music-to-score following (or alignment) is a natural ability of musicians to tell which notes of a given music score are being played in an audio performance. Such systems can be classified as online or offline, depending on whether they can access future audio observations or not. There are numerous applications to online music-to-score alignment, such as automated page turning, and generating synchronized music accompaniments in real-time. We derive and implement exact inference algorithms for fixed-tempo music-to-score alignment systems based on Hidden Markov Models and Hidden Semi-Markov Models. We then evaluate their robustness over synthetic datasets. We study the limits of a fixed-tempo model, the effects of noise on observations, and the impact of pruning the search space.

1 Introduction & Related Work

Music-to-score alignment research dates back to at least 1983 [3], and consists in finding at each time of an audio performance the corresponding position in the score being played. The music score is assumed to be given in a symbolic notation, such as MIDI events or MusicXML files. The audio data can be a recording or a live perfmance, and is typically preprocessed into some spectral representation.

Most algorithms for alignment are based on dynamic programming and amount to warping the audio frames to the score events. Dannenberg [3] used Dynamic Time Warping was to minimize an edit-distance between the audio and the score. Grubb [5] and Raphael [12] introduced probabilistic formulations of the problem, using Hidden Markov Models to represent score events as hidden states and audio frames as the generated observations generated. See [14, 4] for a complete bibliography. More recently, Raphael [13] extends the stochastic formulation by modelling the tempo as a random walk, the event timestamps as the tempo plus some independent gaussian noise, which finally determine the observation distribution of the audio frames, at each time t. The alignments and tempo are then jointly inferred using dynamic programming. Cont [1] uses Hidden Semi-Markov Models to better model the durations of the states as Poisson distributions. Most certainly for efficiency reasons, he jointly uses an Extended Kalman Filter to infer the current tempo from MAP estimates of the score positions, while he uses the predicted tempo to update the duration models. Joder [7] unifies the different approaches as special cases of Conditional Random Fields, and incorporates natural hierarchies of the music score to improve computational properties. Montecchio [8] uses Sequential Monte-Carlo particle filtering for direct approximate inference of the tempo and position. Nakamura [10] returns to a simpler HMM model augmented with skip connections, and proposes a system robust to errors and skips, which extends the applicability to score following for instrument practice.

Offline, Online, and Realtime In the offline setting, both the score and the audio data are available in their entirety, as an audio recording, for instance. Offline music-to-score alignment has applications in music sequencing and mixing, where the result will be stored for later use. On the contrary, online alignment predicts at each audio frame the most probable position in the score. An online algorithm has only access to the past and present audio data, and as such is inherently less accurate, because less information is available. We will show in our experiments that tempo modelling is crucial in the online setting, whereas in the offline systems can get by without no explicit tempo modelling, since the alignment can be constrained to match at both ends. Realtime alignment is online alignment with the constraint that aligning an audio frame should not take longer than the duration of said audio frame. Thus, only algorithms that have time complexity per audio frame independent of the total number of frames can scale to real-time alignment. This is typically not the case of Hidden Semi-Markov Models with general duration models. We will show in Section 4 that in practice we can take advantage of the fast decay of the duration model and prune states during inference, without loosing any noticeable accuracy.

Tempo Modelling Raphael [13] models the tempo process as a random walk with Gaussian increments, in the offline setting. The tempo and alignment are then jointly inferred using dynamic programming and pruning of states.

The Antescofo system, which established the state-of-the-art alignment system as of 2014 [1, 2], is based on a tempo agent and a score agent that bootstrap on each other to jointly decode the tempo and the score position. The beats are are modelled as pulses on a phase circle. The variations in tempo are modelled as an additional pulse which follows a von Mises distribution (Gaussian distribution mapped to a circle). Every time a new score event j is hit, the tempo is re-estimated using an Extended Kalman Filter (tempo agent), based on the current estimated alignment. Reciprocally, the duration models are updated to reflect the change in tempo, and new alignments are predicted with the score agent (HSMM). Even though the system works well in practice, updating the duration models while performing the forward pass introduces implicit dependencies between non-adjacent macro-states. For instance the predicted duration of the first macro-state has an impact on all following macro-states, since it is used by the tempo agent to update the estimated tempo, which will then affect the future duration models. The Markov property is broken, and the computed smoothed probabilties f, f_{out} loose their meaning. Even if in practice, the system is seems stable, there are no guarantees that no divergence will occur since the tempo agent and the score agent bootstrap on each other, and may amplify one another's mistakes.

In our model we make the assumption that the tempo is fixed. The duration models depend only on the nominal duration of the score events. Therefore the duration model of a given note are not affected by the alignments predicted before, which alleviates the need to estimate the current tempo. At the cost of simplicity our system performs exact inference of the score positions. It also avoids the risk of the divergence described above. Our constant-tempo agent has no way to adapt to tempo changes not specified in the score. We perform experiments to see how well the proposed model can perform even when a wrong tempo is assumed, and when the normalized tempo of a performance varies.

2 Our Model

We study the case where a musician performs a piece of music from start to end, without interruption, too many mistakes, or departing from the music score. This corresponds for to a formal performance, as opposed to someone still practising the song, who would go back and forth between different parts of the music score.

2.1 General Framework

Following most music-to-score alignment approaches, we model score events and audio frames as random variables. Scores are represented as a succession of J states, corresponding to positions in the score, as illustrated in Figure 1. Each state j can correspond to a single note, a chord, or a silence, and lives in

$$\mathcal{J} \hat{=} \{0, 1, 2, .., J-1\}$$

The audio frames (x_t) are generally representations directly computed from raw audio data (such as windowed Fourier, MFCC coefficients, pitch estimates, or constant-Q-transform features). They are called the observations, and live in R^d

$$x_0, x_1, x_2, .., x_{T-1} \in R^d$$

A music-to-score alignment is formally a mapping $T \rightarrow \mathcal{J}$, which is represented by states (z_t) taking values in $\mathcal{J}$

$$z_0, z_1, z_3, .., z_{T-1} \in \mathcal{J}$$

We suppose that each audio frame x_t depends only on the current state z_t following an observation model $P(x_t|z_t)$. We then denote

$$b_j(x_t^{t'}) = p(x_t^{t'} | z_t^{t'} = j) \tag{1}$$

Decoding the most probable alignment amounts, in the case of online decoding, to maximizing the posterior probability

$$z \longleftarrow \underset{z}{\text{argmax}}\, p(z_0, .., z_{T-1} | x_0, .., x_{T-1})$$

and for online decoding, to maximizing a each time t the smoothed probability

$$z_t \longleftarrow \underset{z_t}{\text{argmax}}\, p(z_t | x_0, .., x_t) \tag{2}$$

When the process (z_t) is homogeneous, that is, the conditional distributions are time-invariant, then we can define the *occupancy distribution* (see [6])

$$\boxed{d_j(u) \hat{=} P(z_{t+1} \neq j, z_{t-u+1}^t = j | z_{t-u+1} = j, z_{t-u} \neq j)} \tag{3}$$

which is the probability of spending exactly u time steps on state j, and the associated *survival distribution*

$$D_j(u) \hat{=} P(z_{t-u+1}^t = j | z_{t-u+1} = j, z_{t-u} \neq j) \tag{4}$$

$$\boxed{D_j(u) = \sum_{v \geq u} d_j(v)} \tag{5}$$

which is the probability of spending at least u time steps on state j.

In the following sections, we specify the relations between the variables (z_t). For example, if we assume that the musician plays the score in a forward fashion, then all alignments are monotonous. This can be integrated in the generative model by requiring

$$P(z_t \geq z_{t-1} | t < t') = 1$$

on the distribution of states.

2.2 Hidden Markov Model

We use Hidden Markov Models (HMM) as our baseline, because they are simple and have been widely used for speech recognition. The HMM assumes that each state z_t depends only on the previous state z_{t-1} through a transition model $P(z_t|z_{t-1})$. The graphical model is represented on Figure 1. We assume a Linear-Chain HMM, that is, each state j can only transition to itself or to the next score event $j+1$, as shown in Figure 1. The transition probabilities are fully defined by specifying the exit probability

$$p_j = p(z_t = j | z_{t-1} = j-1) \tag{6}$$

The HMM implicitly defines a duration model, specified by its occupancy distribution

$$d_j(u) \hat{=} P(z_{t+1} \neq j, z_{t-u+1}^t = j | z_{t-u+1} = j, z_{t-u} \neq j) \tag{7}$$

$$\boxed{d_j(u) = p_j(1-p_j)^u} \tag{8}$$

which is the probability of staying exactly u frames on state j. We recognize here a geometric distribution. In fact, geometric distributions are the only occupancy distribution that HMMs can represent, which motivates us to investigate HSMMs in section 2.3.

To solve offline decoding, it suffices to apply the Viterbi algorithm [11], which finds the optimal path (z_t) with complexity $O(TJ^2)$.

To solve online decoding, we explore two approaches. The first approach is at each time t, to optimize for the optimal path $z_0, .., z_t$, and output z_t, which corresponds

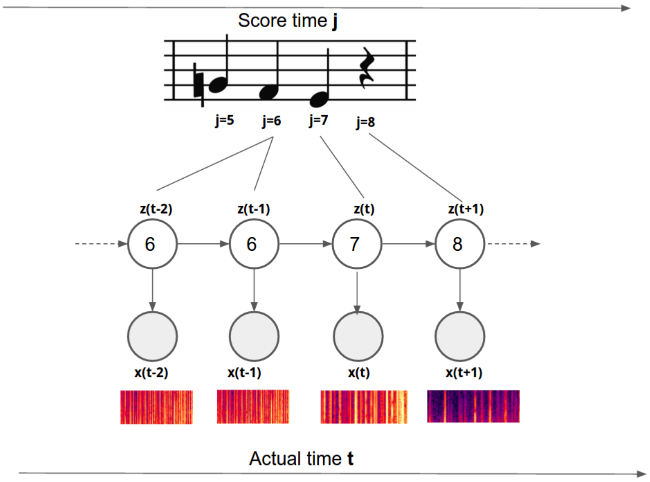

Figure 1: HMM Bayesian Network

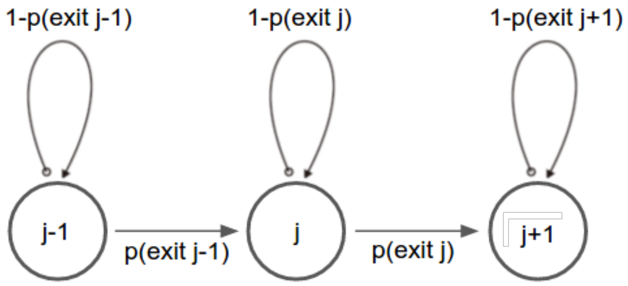

Figure 2: HMM Transition Probability

conceptually to applying the Viterbi algorithm to each observed subsequence

$$z_t \longleftarrow \operatorname*{argmax}_{z_t} \max_{z_0,..,z_{t-1}} p(z_0,..,z_t|x_0,..,x_t)$$

The second approach consists in directly optimizing the marginal probability of each state, given the current observations

$$z_t \longleftarrow \operatorname*{argmax}_{z_t} p(z_t|x_0,..,x_t)$$

which is equivalent to applying the forward-backward algorithm (without the backward pass) on each subsequence.

2.3 Hidden Semi-Markov Model

Following [1], we decide to model the duration model $d_j(u)$ explicitly, using Hidden Semi-Markov Models (HSMM). Hidden Semi-Markov Models [9] combine the Markov states j with a duration variable $u \in \mathbb{N}$ into a *macro-state* (j,u). The macro-states can then be made to follow any arbitrary occupancy distribution $d_j(u)$ (as defined in Equation 3). Figure 3 represents the transitions between macro-states.

We now derive the right-censored Forward algorithm as described in [6, 2], and specialize it to our framework. We introduce the smoothed probabilities, which we will later maximize on z_t

$$f(t,j) \hat{=} p(z_t=j,x_0^t) \propto p(z_t=j|x_0^t) \tag{9}$$

and the out-of-state probability

$$\begin{aligned} f_{out}(t,j) &\hat{=} p(z_{t+1}\neq j, z_t=j, x_0^t) \qquad (10)\\ &= p(z_{t+1}=j+1, z_t=j, x_0^t) \qquad (11)\end{aligned}$$

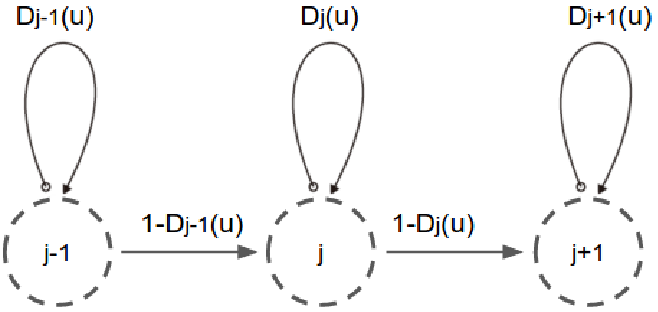

Figure 3: HSMM Transition Probability

which is the probability of exiting macro-state j for another macro-state. Because we constrain the alignment to be monotonic, the next distinct macro-state after j can only be $j+1$, hence Equation 11.

Marginalizing over all possible durations u of the previous macro-state, we derive[1] the smoothed probabilities

$$\begin{aligned} f(t,j) \hat{=}& p(z_t=j,x_0^t)\\ =& \sum_{1\leq u\leq t+1} p(x_0^t, z_{t-u+1}^t=j, z_{t-u}=j-1)\\ =& \sum_{1\leq u\leq t+1} p(x_{t-u+1}^t, z_{t-u+1}^t=j, z_{t-u}=j-1, x_0^{t-u})\\ =& \sum_{1\leq u\leq t+1} p(x_{t-u+1}^t=j|z_{t-u+1}^t=j)\\ &\times p(z_{t-u+1}^t=j|z_{t-u}=j-1)\\ &\times p(z_{t-u+1}=j, z_{t-u}=j-1, x_0^{t-u})\end{aligned}$$

Similarly, we derive the exit probabilities

$$\begin{aligned} f_{out}(t,j) \hat{=}& p(z_{t+1}=j+1, z_t=j, x_0^t)\\ =& \sum_{1\leq u\leq t+1} p(x_0^t, z_{t+1}=j+1, z_{t-u+1}^t=j,\\ &\qquad z_{t-u}=j-1)\\ =& \sum_{1\leq u\leq t+1} p(x_{t-u+1}^t, z_{t+1}=j+1, z_{t-u+1}^t=j,\\ &\qquad z_{t-u}=j-1, x_0^{t-u})\\ =& \sum_{1\leq u\leq t+1} p(x_{t-u+1}^t=j|z_{t-u+1}^t=j)\\ &\times p(z_{t+1}=j+1, z_{t-u+1}^t=j|z_{t-u}=j-1)\\ &\times p(z_{t-u+1}=j, z_{t-u}=j-1, x_0^{t-u})\end{aligned}$$

We obtain the following recursions all $j \in \mathcal{J}$ and $t \geq 1$

$$\boxed{\begin{aligned} f(t,j) &= \sum_{1\leq u\leq t+1} b_j(x_{t-u+1}^t) D_j(u) f_{out}(t-u, j-1)\\ f_{out}(t,j) &= \sum_{1\leq u\leq t+1} b_j(x_{t-u+1}^t) d_j(u) f_{out}(t-u, j-1)\end{aligned}}$$

(12)

(13)

[1] We take notations similar to [2], but we define f and f_{out} as joint probabilities with x_0^t, instead of probabilities conditioned on x_0^t

with initialization

$$\boxed{\begin{aligned} f_{out}(0, j = 0) &= d_0(1) \qquad (14)\\ f_{out}(0, j \neq 0) &= 0 \qquad (15)\end{aligned}}$$

We then decode by maximizing at each frame t

$$z_t \longleftarrow \underset{j}{\operatorname{argmax}}\, f(t, j) \tag{16}$$

3 Discussion

Monotonicity of alignment Note that the online decoding scheme based on maximizing the smoothed criterion that we propose does not guarantee a monotonic alignment in real-time. If our goal is really to predict the most probable position at every time step, there is no particular reason that the resulting alignment would be monotonic. However this can be problematic when one wants to play a pre-recorded music accompaniment along. It can also be a problem in automated page turning, because we may want to avoid jerking back and forth between pages. Instead, one could imagine recursively optimizing the criterion

$$f(t, j)1_{j \in \{z_{t-1}, z_{t-1}\}} = p(z_t = j, z_{t-1} | x_0^t) \tag{17}$$

equivalent to

$$z_t \longleftarrow \underset{j \in \{z_{t-1}, z_{t-1}\}}{\operatorname{argmax}} \quad f(t, j) \tag{18}$$

but then inference would be less robust, since as soon as a state j is reached, there is would be no way to predict states $j' < j$, resulting in overestimated alignments.

4 Experiments

For our purposes, we perform several simplifications. We assume that the music is played in a forward fashion, from beginning to end, without skipping parts. We can assume without loss of generality that the reference music score has a fixed tempo (bpm), because we can hard-code tempo changes simply by scaling the duration of score events by an appropriate factor.

We also adopt a very simple observation model: a unimodal Gaussian centered around the score event pitch

$$x_t | z_t \sim \mathcal{N}(pitch(z_t), \sigma^2) \tag{19}$$

where the score is assumed to be monophonic. Finally, we suppose that there are no blanks, although the current model can be easily extended by appending a velocity (volume) scalar to the pitch. Note that the HMM and HSMM decoding algorithms derived above do not assume a particular form of the observation model, so it is fine to experiment with more realistic ones later.

In the following sections, we compare HMM to HSMM models. We study the robustness to different types of variability in music scores and their interpretation. We then analyze the effects of various parameters.

Metrics for evaluation We evaluate the quality of an alignment using the average absolute offset in seconds, and the percentage of absolute offsets higher than 100 ms, which we will simply refer to as the *error*.

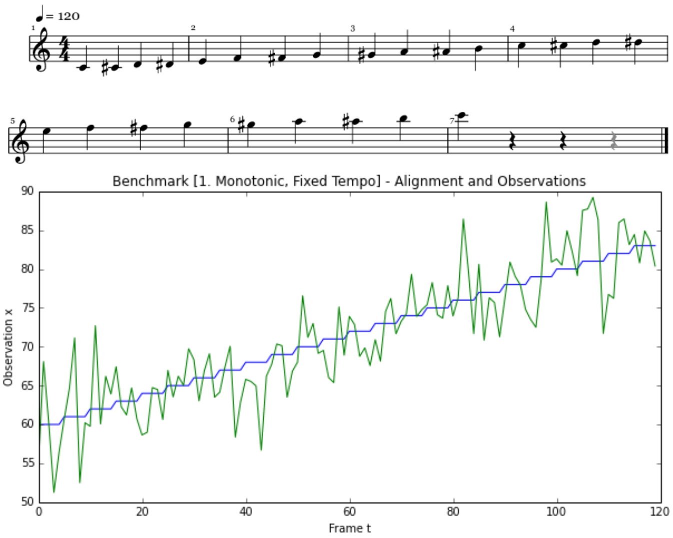

Figure 4: **top**: Simple Monotonic Pattern **bottom**: Observations (blue is actual pitch, green is noisy observation).

Default values By default, we match the mean of the occupancy distributions with the nominative durations the associated score events . We keep the same observations models for generation of the synthetic dataset, and decoding (same σ^2).

4.1 Simple Monotonic Pattern

We first test the models on a simple monotonic pattern: a sequence of notes with pitch increments of a halftone, over two octaves. Recall that there are 12 logarithmically-space halftones per octave, and an octave corresponds to doubling the frequency. The pattern is represented on Figure 4.

The alignments given by maximizing the smoothed probability are given in Figures 5,**??**. In that case, the HMM and HSMM perform similarly well, with a 1.6% error against 0%. In that case, using a HSMM doesn't improve significantly over the HMM.

4.2 Repeated Notes

In this second experiment we consider a folk song with multiple, possibly consecutive occurrences of the same notes, given on Figure 6. Dealing with repetition requires tracking of the states (both done by HMM and HSMM), and correctly aligning consecutive instances of the same note require some tempo modelling (done by HMM and HSMM with geometric and Poisson occupancies).

Here the online HSMM achieves 4.5% error, against 16% for the online HMM, even outperforming the 13% error for the offline HMM. Alignments are shown in Figure 7. Also note that the offline HMM tends to take make jumps a bit ahead in predicted position, instead of smoothly ascending like the HSMM. This is likely a consequence of the strong weight put by the geometric occupancy distribution on small durations u, with effects exaggerated on consecutive occurrences (such as the repeated F and E in measure 2, and the repeated D at the end of measures 1 and 3).

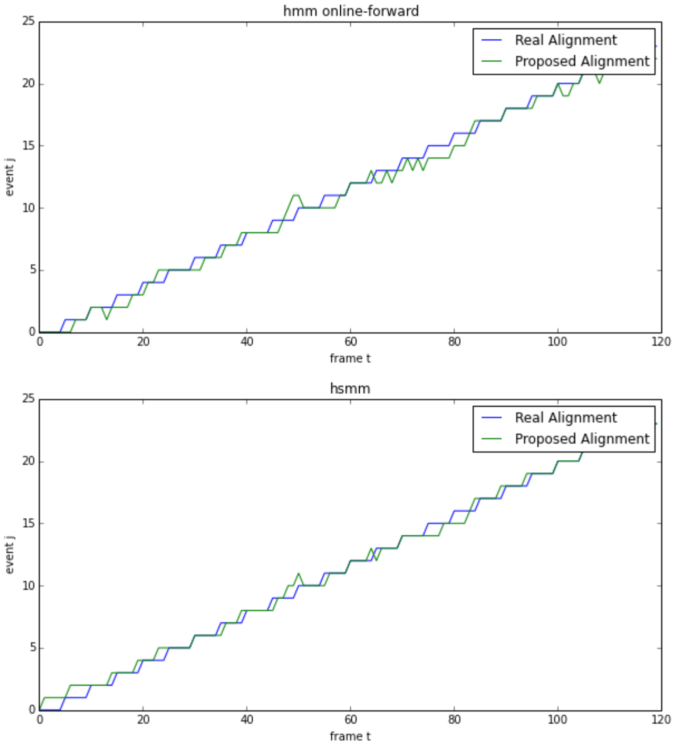

Figure 5: Simple Monotonic Pattern - **top:** HMM, **bottom:** HSMM

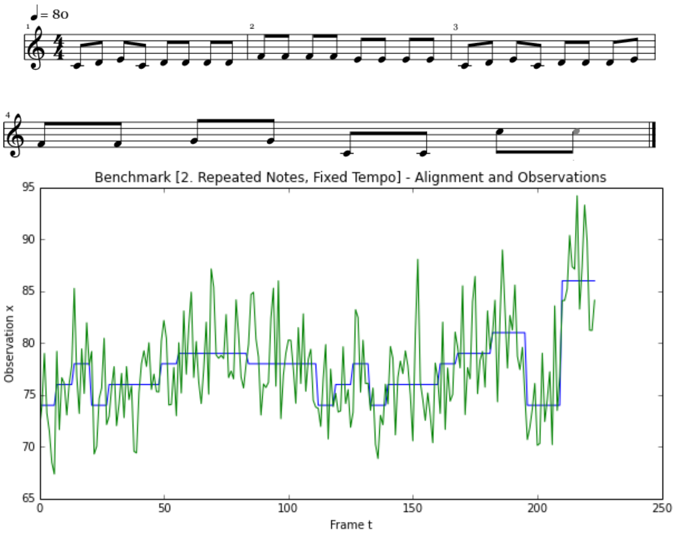

Figure 6: **top**: Repeated Notes (folk song) **bottom**: Observations (blue is actual pitch, green is noisy observation).

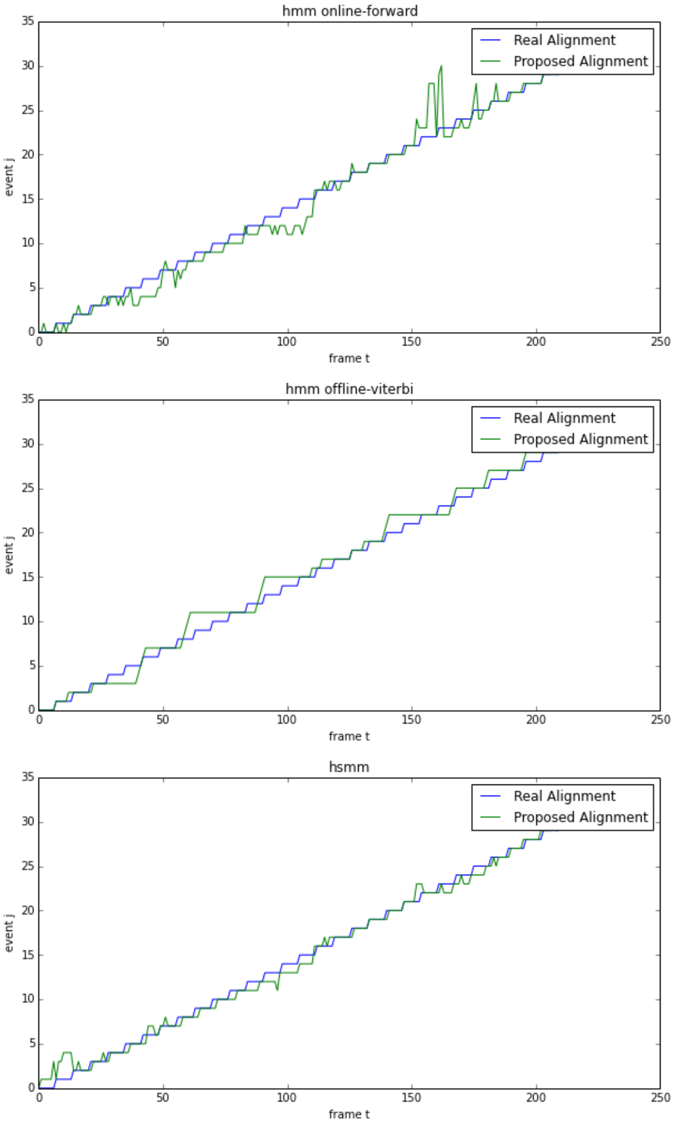

Figure 7: Repeated Notes - **top:** HMM online, **middle**: HMM offline, **bottom**: HSMM online

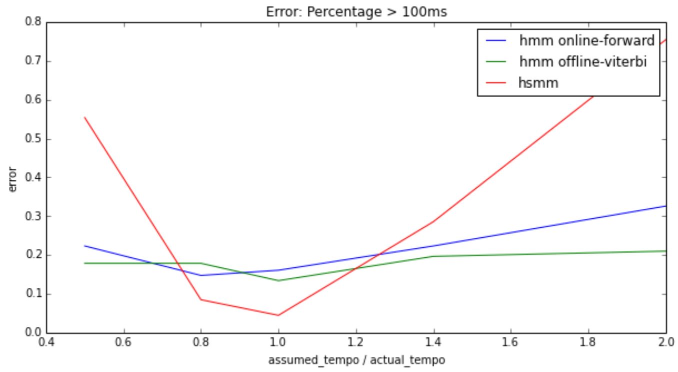

Figure 8: Effects of Biased Tempo Assumption

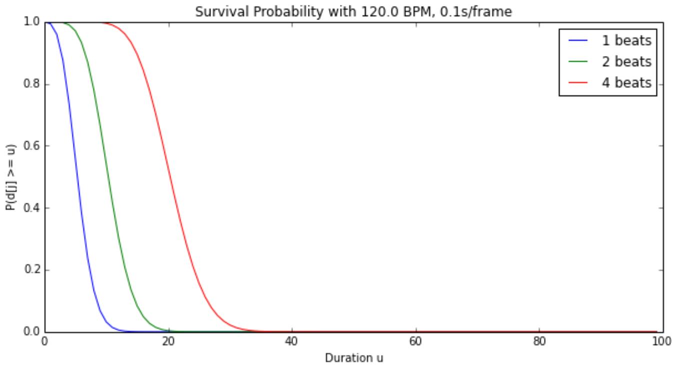

Figure 9: Pruning states - Survival distribution for Poisson

4.3 Biased Duration Model

Because the HMM and HSMM duration models require the tempo to be specified a priori, we study their robustness to a wrong tempo assumption. We evaluate offline viterbi for HMMs, online forward decoding of HMMs, and online decoding of HSMMs, against under/over-estimated tempos. The results plotted in Figure 8 show that offline Viterbi is the most robust to a wrong duration model, with almost constant error across assumed tempo. The forward algorithm for HMMs is also fairly robust to a wrong tempo. Unfortunately the HSMM is more sensitive to a wrong tempo, and the error increases rapidly, with performance worse than the HMM for

$$\frac{assumedTempo}{actualTempo} \notin (0.8, 1.2)$$

4.4 Pruning of states

The HMM online and offline decoding algorithms have total decoding time $O(TJ^2)$ which can be reduced to $O(TJ)$ by taking advantage of the linear-chain structure of the states, which corresponds to $O(J)$ at each online time-step t.

The HSMM right-censored forward algorithm has total decoding time $O(T^2J^2)$, which becomes $O(T^2J)$ from the linear-chain structure. It can be further reduced to $O(TJU)$ where for $u \geq U$ the survival distribution is negligible $D_j(U) \ll 1$. In practice if the longest event lasts 4 beats, at 120 BPM and 100 ms/frame, then $U = 50$ is fine, as shown in Figure 9. The complexity is then $O(JU)$ at each online time-step, which is about $U \sim 50$ times slower than HMM, but fast enough for real-time decoding with an efficient implementation.

4.5 Extending to actual audio data

We attempt to align a live performance of Chopin's *Fantaisie Impromptu* by Yundi Li with a MIDI rendering of the song. The Viterbi and Forward algorithms can be applied for any observation model. Thus we compute their Constant-Q-Transforms, which is a representation commonly used in music-related signal processing because the frequency bins are logarithmically spaced and reflect the natural spacing of tons. Note the visual similarity between the reference CQT (rendered MIDI) and the candidate CQT in Figure 10. The candidate CQT looks smoother which we may attribute to the use of the sustain pedal, whereas the MIDI file was rendered with no such effects.

We then compute pairwise cosine distances between the spectra of the two performances. In that distance matrix (Figure 11) each path from the top-left to the bottom-right corresponds to a possible alignment. The optimal path appears clearly in dark blue. It is roughly piecewise-linear, with 3 apparent segments. Since the MIDI file was rendered with no tempo changes, it is likely that the line segments correspond to 3 music sections with different but steady *tempi*. Moreover, several lines appear parallel to the optimal path. By listening to the recordings, they identify to repetitions contained in the original music score.

Possible observation probabilities are multivariate Gaussians centered around the spectra of the reference sequence. We then try to apply the algorithms presented in the previous section, but it is not as straightforward as one may think. Indeed, we render the MIDI score to audio and consider each resulting CQT frame to be one score event, whereas in [1] the score events correspond to the notes and chords that a musician would normally read. Our approach is simpler in practice since we can use any third-party MIDI synthesizer to get the CQT spectrum, instead of generating/storing prototypes for each possible chord. The downside is that there are many more score events, which is problematic because of the quadratic dependence of the decoding algorithms on J the number of score events. The answer is to perform a beam-search on alignments: at each time t, only keep the H most probable events and discard the rest. Then the complexity becomes $O(THU)$ and decoding becomes tractable in real-time. We will test that approach and evaluate the influence of H in the future.

5 Acknowledgements

Thanks to Simon Lacoste-Julien, Guillaume Obozinski, and Francis Bach for teaching an excellent introduction to probabilistic graphical models, and making me want to learn more on Bayesian statistics. Special thanks to Philippe Cuvillier and Arshia Cont for having me over at IRCAM to answer all my questions about the model and implementation details.

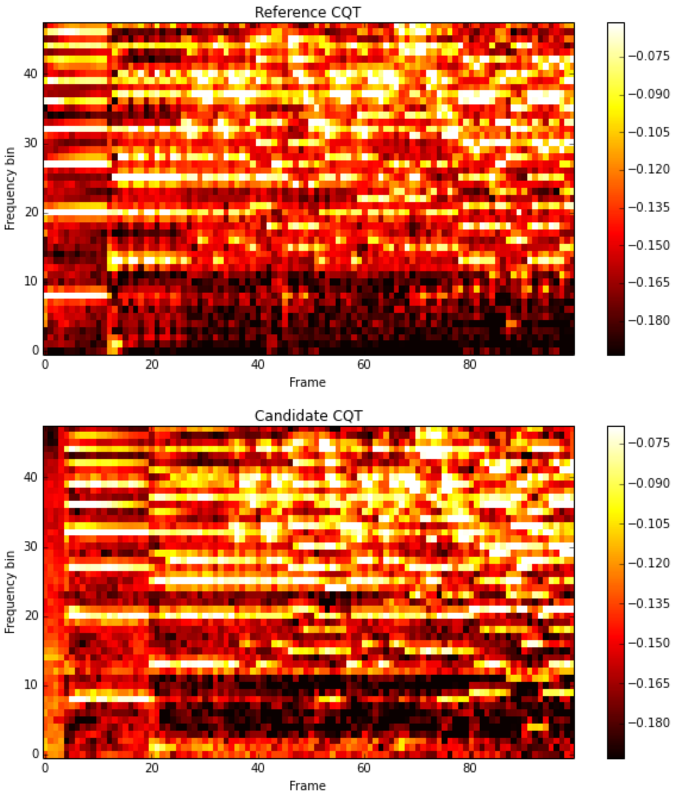

Figure 10: Constant-Q-Transforms. **top:** reference MIDI, **bottom** candidate live performance (top).

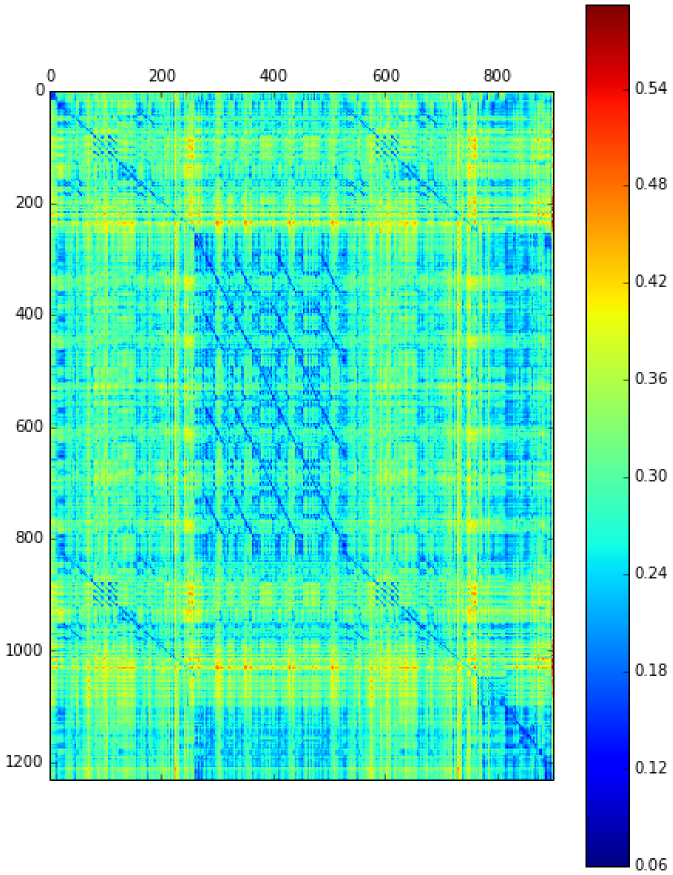

Figure 11: Distance matrix between real performance and MIDI rendering.

6 Conclusion and Future Work

We have presented and proved correctness for two music-to-score alignment systems with fixed-tempo modelling. The first baseline system is a HMM (online and offline) which is faster and simpler, but less accurate. The second system is a HSMM (online, although offline is possible too), which is slower and more complex, but has more descriptive power because it can have any arbitrary duration model. We compared the models and evaluated their robustness over synthetic datasets with various challenging features. We concluded that HSMMs perform generally better than HMMs, even offline, but only under the condition that the assumed tempo is close enough to the performance tempo. We have studied the applicability of the presented algorithms to actual audio data and concluded that a beam-search might be necessary for realtime applications.

The next step is to investigate models for variable tempo. The goal is to have comparable performance with the audio agent of [1] based on Extended Kalman Filters, while maintaining a fully probabilistic model as in [13] where the tempo process is modelled as a random walk.

The code is available at `github.com/gabrielhuang/tmalign`.

References

[1] Arshia Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):974–987, 2010.

[2] Philippe Cuvillier and Arshia Cont. Coherent time modeling of semi-markov models with application to real-time audio-to-score alignment. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, pages 1–6. IEEE, 2014.

[3] Roger B Dannenberg. *An on-line algorithm for real-time accompaniment.* Ann Arbor, MI: MPublishing, University of Michigan Library, 1984.

[4] Roger B Dannenberg and Christopher Raphael. Music score alignment and computer accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.

[5] Lorin Grubb and Roger B Dannenberg. A stochastic method of tracking a vocal performer. In *Proceedings of the ICMC*, pages 301–308, 1997.

[6] Yann Guédon. Hidden hybrid markov/semi-markov chains. *Computational statistics & Data analysis*, 49(3):663–688, 2005.

[7] Cyril Joder, Slim Essid, and Gaël Richard. A conditional random field framework for robust and scalable audio-to-score matching. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(8):2385–2397, 2011.

[8] Nicola Montecchio and Arshia Cont. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques. In *Acoustics, Speech and Signal Processing*

(ICASSP), 2011 IEEE International Conference on, pages 193–196. IEEE, 2011.

[9] Kevin P Murphy. Hidden semi-markov models (hsmms). *unpublished notes*, 2, 2002.

[10] Tomohiko Nakamura, Eita Nakamura, and Shigeki Sagayama. Acoustic score following to musical performance with errors and arbitrary repeats and skips for automatic accompaniment. *Proceedings of Sound and Music Computing*, pages 299–304, 2013.

[11] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[12] Christopher Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(4):360–370, 1999.

[13] Christopher Raphael. A hybrid graphical model for aligning polyphonic audio with musical scores. In *ISMIR*, 2004.

[14] Diemo Schwarz. Score following commented bibliography, 2003.