HOW TO BE LOST: PRINCIPLED PRIMING AND PRUNING WITH PARTICLES IN SCORE FOLLOWING

Charles Fox

Robotics Research Group
Dept. Engineering Science
University of Oxford
charles@robots.ox.ac.uk

John Quinn

Institute for Adaptive and
Neural Computation
School of Informatics
University of Edinburgh

ABSTRACT

Previous work in score following has provided methods for aligning a skilled live performance to a symbolic or audio score. In the Bayesian framework, ideal generative models require $O(n)$ computations at each real time step where n is the length of the score. In practice, heuristic thresholds have been used to consider only a subspace of generative models with high priors conditioned on the previous state. These heuristics work well for skilled performances but fail when large errors are made by amateur musicians. We present a novel Priming Particle Filter for audio scores which places the order-limiting heuristic on a firm foundation and adds the ability to recover from large errors by using psychologically-inspired bottom-up priming in addition to regular sequential importance sampling.

1. INTRODUCTION

Automated score following is the task of listening to a human performance of a predetermined score and playing a predetermined accompaniment in real-time which adapts to tempo changes and errors of the player.

Dannenberg [1] applied Dynamic Time Warping (DTW) to follow MIDI performances, allowing for insertion and omission of notes. The DTW path gradient is used to approximate tempo and hence provide accompaniment. In real-time, the optimal path is noted at each step and the search on the next row is constrained to nearby elements within an arbitrary window. Hence the complexity at each time point is $O(1)$ rather than $O(n)$ where n is the score length. This approach was extended to audio alignment: score and observation audio are cut into frames, and 'chroma' vectors are extracted which shift all pitches to within a single octave. A distance measure is defined on pairs of chroma vectors, and used as the local distance in DTW.

DTW is a likelihood-based approach to alignment. To perform it exactly requires each incoming live observation to be dot-producted with every score position, then the shortest path through this matrix is found. There is no explicit prior over possible paths: we can consider it flat. In practice, it is intractable to perform comparisons with the whole score, so DTW algorithms typically consider only a fixed window (of perhaps 100 audio frames) about the previous best point. This can be thought of as specifying a finite plateau transition prior centered around the previous MAP point. Importantly, this heuristic specifies not only a prior belief but also a *computational* simplification: it tells us to *consider* only those hypotheses in the plateau rather than all possible generating score states.

Moving from the plateau prior to an arbitrary state transition matrix, DTW becomes a general Hidden Markov Model. Rather than use score audio frames as states (as in DTW), score-following HMM researchers have tended to use a reduced model of the score, with one or several states per note ([2], [3]). Transitions are typically allowed between consecutive score states, self-transitions, and jumps to the end of the note, and their parameters fitted to previous performances.

While HMMs provide explicit prior beliefs about transitions, they do not address the computational problem of which beliefs to *consider* in computations. As in DTW, practical HMMs generally apply some kind of cutoff threshold to limit the space of score states under consideration. For example, Raphael [5] uses a heuristic which prunes all but a fixed number of most probable posterior hypotheses at each time step. These states tend to be closely clustered. This works well for Raphael's intended users: professional performers who are unlikely to make large errors, and whose small-scale tempo deviations from previous performances can be tracked with high accuracy. But for amateurs who may make very large leaps around the piece it runs the risk of becoming irrecoverably lost.

Heuristic hypothesis management is the 'dirty secret' of much of Bayesian inference. Particle Filtering [8] is a more principled approach to hypothesis management in dynamic Bayesian networks including HMMs. Rather than keeping the most probable hypotheses, it samples from them at each time step. This can allow for less probable paths to be explored in the hope that they may eventually provide a better global path than locally probable ones. Particle filtering has been applied to model-based score-following HMMs by [7]. This allows for larger leaps in the score than a 'take only the best' approach, but it is still possible for particle filters to become lost ('diverge') and they are then unlikely to recover.

Human musicians do not just rely on a prior information to select location hypotheses to consider: hypotheses can also be *primed* [9] bottom-up from observed features. For example a song may have a distinctive chord change at the start of a the chorus: when this change is heard, it primes us to consider that we might be at that location – even if our prior beliefs were focused elsewhere. Within the framework of particle filtering, we define priming to be the injection of new samples into the system driven only by bottom-up features, regardless of their priors. We present a novel 'Priming Particle Filter' (PPF) which uses this technique to recover from being lost by recognizing and allowing a probability for its own error, then creating new hypotheses based on bottom-up note changes. Unlike existing HMM score-followers, we have returned to the DTW-style audio-based score, and avoid the need for explicit note-based score models. To demonstrate the priming particle filter we have used relatively simple features for likelihood computations, but we suggest that the PPF could be a useful addition to all state-of-the-art score followers ([4], [6]) with more advanced features.

2. PARTICLE FILTERING IN HMMS

Hidden Markov Models assume a discrete hidden state $x_i[t]$ at discrete time steps t with linear Markovian transitions, $P(x_i[t+1]) = \sum_j P(x_i[t+1]|x_j[t])P(x_j[t])$, together with observations $P(y[t]) = f(x[t])$. The filtering posterior at each step is given recursively by $P(x_i[t]|y[1:t]) = \frac{1}{Z}P(y[t]|x_i)\sum_j P(x_i[t]|x_j[t-1])P(x_j|y[1:t-1])$ where the sum is over all possible hidden states, so scales with the length of a score. Particle Filters can be used to approximate the sum by maintaining a limited set of samples from each $P(x[t]|y[1:t])$. We base our Priming Particle Filter on a standard particle filtering technique, Sequential Importance Resampling, whose algorithm is:

for each time step t **do**
  for $p = 1 : N$ **do**
    sample $x^p[t] \sim P(x_i|x^p[t-1])$
  end for
  for $p = 1 : N$ **do**
    $w^p[t] := \frac{1}{Z}w^p[t]P(y[t]|x^p[t])$
  end for
  if $\frac{1}{\sum_{p=1}^{N}(w^p)^2} < N_{thresh}$ **then**
    resample $x^p[t] \sim P(x[t] = i) = w^i[t]$
  end if
end for

where N is the number of particles and N_{thresh} is a hand-set threshold for number of effective particles. The resampling step becomes necessary in the case of degeneracy (i.e. when most particle weights $w^p[t]$ become small). See [8] for details.

3. SCORE FOLLOWING MODEL

3.1. Tempo model

At each (discrete) frame of live performance time $t^{(l)}$, the hidden state is the score position in frames, $t^{(s)}$ and the current tempo. These hidden state variables are modeled as continuous and are lazily discretized only when a hard output frame decision is required. We write the score time as a function of the live time, $t^{(s)}[t^{(l)}]$, and write the current tempo as $\dot{t}^{(s)}[t^{(l)}]$. The hidden state evolves as the damped switching stochastic process:

$$t^{(s)}[t^{(l)}] = t^{(s)}[t^{(l)} - 1] + \dot{t}^{(s)}[t^{(l)} - 1] + \epsilon_{t^{(s)}}$$
$$\dot{t}^{(s)}[t^{(l)}] = \rho(\dot{t}^{(s)}[t^{(l)} - 1] + \epsilon_{\dot{t}^{(s)}} - 1) + 1$$

where ρ is a damping coefficient and $\epsilon_{t^{(s)}}$ are from the Gaussian $N(0, \sigma_{t^{(s)}})$ and $\epsilon_{\dot{t}^{(s)}}$ are from the switching mixture of Gaussians $\alpha_1 N(0, \sigma_{\dot{t}^{(s)}}) + \alpha_2 N(0, \sigma_{\dot{t}^{(s)}})$ with $\alpha_1 + \alpha_2 = 1$. The two mixture components model a small tempo drift due to player or tracking errors and a separate large tempo change due to performer style changes. A mixture is used to encourage occasional, sudden large changes (e.g. at new phrases) whilst discouraging gradual large tempo changes (e.g. during notes). We do not perform any learning of parameters – they are set by hand. Particles contain the 2-element state $(t^{(s)}[t^{(l)}], \dot{t}^{(s)}[t^{(l)}])$, i.e. a position and speed in the score.

3.2. Likelihoods

Likelihoods are approximated using chroma features similar to [1]. Offline score $y^{(s)}$ and online live $y^{(l)}$ audio is sampled at 2756.3Hz (=44.1kHz/16) and cut into 512-point frames $y[t]$ with 448 point overlaps. The signals are differentiated to remove linear components then Hann-windowed and the power FFTs are computed. The total energy $e[t]$ and normalized power FFT $Y[t]$ are stored:

$$Y[t] = \frac{1}{Z}|FFT(Hann(y'[t]))|^2$$
$$e[t] = \sum_{\omega} |(FFT(Hann(y'[t]))|^2$$

where Z is a normalizing constant and ω sums over frequency components. A filterbank of note detectors is projected onto the power FFT with each detector having a 10-peaked harmonic series:

$$n_i[\omega] = \frac{1}{Z}\sum_{k=1}^{10} \exp(-k)\Phi(k\omega; 2^{\frac{i}{12}}\omega_0, \sigma)$$

where $\Phi(\omega; \mu, \sigma)$ is the Gaussian pdf, ω_0 is the frequency of the lowest considered pitch, and σ is chosen by hand to give a reasonable spread but without interfering with neighboring frequencies. (Ideally σ would be derived from acoustic theory or fit to data.)

Note detectors over a two-octave range are summed to give a 12-valued octave-independent chromacity vector:

$$N_i(Y[t]) = \sum_j \langle n_{i+12*j}|Y[t]\rangle$$

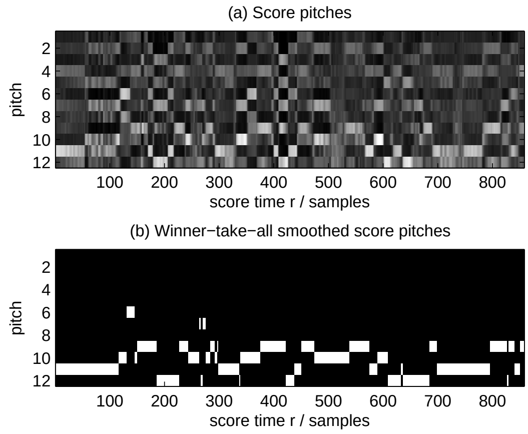

Figure 1. Chroma and change points of score performance

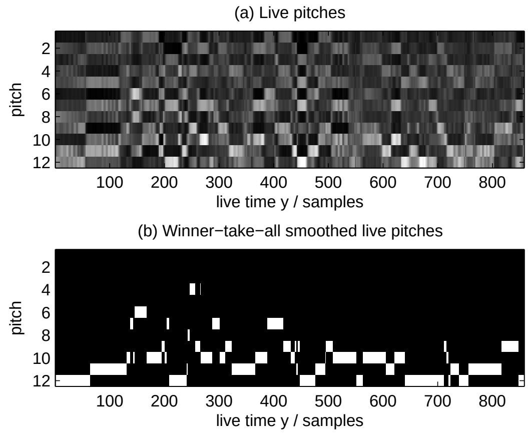

Figure 2. Chroma and change points of live performance

We use an inner product between chroma vectors as an approximate spectral likelihood (i.e. the probability of the observed spectrum at live time $t^{(l)}$ given score time $t^{(s)}$):

$$\lambda_s(Y^{(s)}[t^{(s)}], Y^{(l)}[t^{(l)}]) = \frac{\langle Y^{(s)}[t^{(s)}] | Y^{(l)}[t^{(l)}] \rangle}{\sqrt{e^{(s)}[t^{(s)}] e^{(l)}[t^{(l)}]}}$$

Energy values are normalized by local windowed (length L) mean and variance to compute novelty features:

$$v[t] = \frac{e[t] - \mu_{j=t-L:t}(e[j])}{\sigma_{j=t-L:t} e[j]}$$

We use a Gaussian approximation of energy likelihoods:

$$\lambda_v(v^{(s)}, v^{(l)}) = \frac{1}{Z} \exp \frac{-(v^{(s)} - v^{(l)})^2}{2\sigma_v^2}$$

and assume that chroma distance and energy are sufficient statistics for the total likelihood:

$$P(t^{(l)} | t^{(s)}) \approx \lambda_s(Y^{(s)}[t^{(s)}], Y^{(l)}[t^{(l)}]) \lambda_v(v^{(s)}[t^{(s)}], v^{(l)}[t^{(l)}])$$

3.3. Injecting primed particles

A fundamental assumption in Bayesian inference is that we posses the exhaustive set of hypotheses $\{H_i\}$ to explain some data D. It is this exhaustiveness that allows us to convert likelihoods into posterior probabilities. However in realtime score following we do not have computational resources to consider all possible score position hypotheses, we must consider only a subset of them. Previous models have used heuristics to choose this subset: we (with [7]) use sequential importance samples (particles) to make this choice in a principled way. The standard particle filter assumes that the particles form the exhaustive hypothesis set at each step. But this does not allow the model to represent the possibility of its own failure: in practice it is possible for a particle filter to get lost, and for the best hypothesis (and even the whole area around it) to be excluded from consideration. But the particle filter is blind to this. What we would like to do is compute the probability that all particles have failed, $P(lost)$, and treat 'being lost' as an alternative explanation. When we know we are lost we will then consider a number of newly injected particles $\{H_j^*\}$ based on bottom-up priming, with

$$P(H_j^* | D, C) = P(lost | D, C) P(H_j^* | lost, D, C)$$

$$P(H_j^* | lost, D, C) = \frac{1}{Z} P'(H_j^* | feature(D)) P'(H^* | C)$$

where C is the context (i.e. the previous state for dynamic networks like HMMs), $feature(D)$ is some bottom-up priming feature computed from the data, and P' are probability *factors*. In practice, all of the above terms are intractable (in the sense that computing them would require an exact solution of the very inference problem that the particle filter is approximating) but can be approximated from domain knowledge or historical statistics.

We estimate $P(lost | D, C)$ by looking at a windowed running average data likelihood. We use an exponentially weighted moving average of $P(D[t] | \hat{H}[t])$, the evidence of the MAP hypothesis at each step. When this average falls below a threshold we consider that we might be lost with a fixed probability. In practice we approximate this by removing the worst m particles from the set and replacing them with newly primed particles, where m is the number of these primes. (More detailed methods could model how probable being lost is given how far below threshold we are, and learn this model from historical data.)

$P'(H_j^* | feature[D])$ is approximated by a simple bottom-up priming scheme. We use sparse Boolean features and assume that all score states having the feature are equally likely given the presence of the feature when lost. For our score-following task, we use changes in the maximum windowed running average pitch as features. These *roughly* correspond to onsets of new notes. We maintain a cache of known positions of these features in the score, and consider these positions as hypotheses when the same features are found in a lost live performance. Speeds are created for these particles assuming that the tempo has been constant since the point of becoming lost.

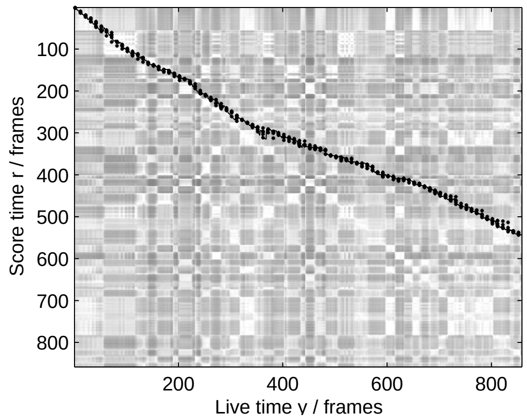

Figure 3. Alignment without priming.

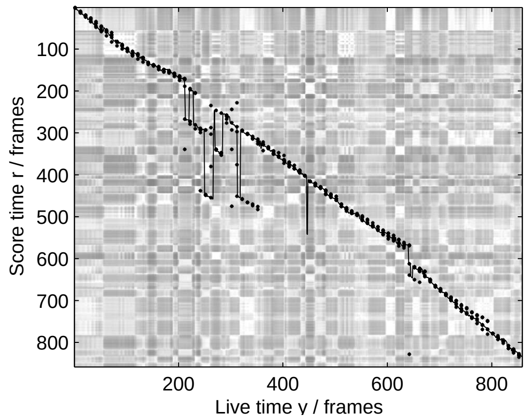

Figure 4. Alignment with priming.

$P'(H^*|C)$ can be approximated in score-following by considering how far removed the primed hypothesis H* is from where the particle filter's previous MAP estimate was, and penalizing very large jumps. Our system uses a simple plateau function to allow only primes within a fixed distance of the previous MAP.

Our method corresponds to altering the distribution from which particles are sampled from in the SIR algorithm. If we know that we are lost, prior knowledge about likely 'primed' positions is used to obtain a more appropriate set of samples.'

4. RESULTS

We demonstrate the priming particle filter on the first seven bars of the *Andante Cantable* from the Rimsky-Korsakov *Trombone Concerto*. Recall that no symbolic score is required: an audio recording is used as the score. Fig. 1 shows the chroma vectors and priming features from the audio score; fig. 2 shows the same for the live recording.

The live recording contains a short error around frame 200. Running the standard particle filter with 5 particles results in getting lost at this point as shown in fig. 3. The figure shows the complete spectral similarity matrix in the background, in which the central white diagonal stripe is the true path. The particles are plotted as dots and the MAP particle path is drawn as a line. Fig. 4 shows the same example running under the Priming Particle Filter. The large jumps are where lostness recognition and priming has occurred. It can be seen that the filter gets lost several times but is able to recover every time.

5. DISCUSSION

We have demonstrated the Priming Particle Filter on a simple example for illustration purposes only, and running with more than 5 particles on this example generally avoids getting lost in the first place. Our demonstration PPF is a relatively simple score follower using very basic likelihood features and a Markov hidden state in contrast to state-of-the-art score followers such as [4], [6] which have more advanced features and higher-level state models. However we suggest that all state-of-the-art models could benefit from the simple addition of a PPF to allow them to recover from being lost when used by non-professional performers.

6. ACKNOWLEDGMENTS

Thanks to Matthew Dovey, Chris Raphael and Diemo Schwarz for discussions.

7. REFERENCES

[1] R. Dannenberg and N. Hu. "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors" *ICMC*. 2003.

[2] C. Raphael. "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models" *IEEE PAMI*. 21(4):360-370. 1999.

[3] N. Orio and F. Dechelle. "Score Following Using Spectral Analysis and Hidden Markov Models" *ICMC*. 2001.

[4] C. Raphael. "Music Plus One: A System for Flexible and Expressive Musical Accompaniment" *ICMC*. 2001.

[5] C. Raphael. "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models" *IEEE PAMI*. 21(4):360–370, 1999.

[6] A. Cont and D. Schwarz. "Score Following at Ircam" *MIREX Score Following Contest*, 2006.

[7] A. Cont. "Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative constraints and Hierarchical HMMs" *IEEE Int. Conf. Acoustics and Speech Sig. Proc.*, 2006.

[8] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice* Springer. 2001.

[9] C. Wiggs and A. Martin. "Properties and mechanisms of perceptual priming." *Cur. Op. Neurobiology (CNS). (8) 227-233*. 1998.