Bayesian Real-time Adaptation for Interactive Performance Systems

Ali Taylan Cemgil and Bert Kappen
SNN, University of Nijmegen, The Netherlands
email: {taylan,bert}@mbfys.kun.nl

Abstract

We introduce Bayesian online learning for real time parameter adaptation on a tempo tracking task. We employ a variational extension of the Expectation Maximization algorithm for online parameter estimation. Simulation results on a real dataset indicate that online adaptation has the potential of capturing performer specific features in realtime.

1 Introduction

An interactive music performance system (IMPS) (Rowe, 1993) is a computer program that "listens" to the actions of a performer and generates responses in real-time. IMPS applications include (but are not limited to) automatic accompaniment or improvisation. One important goal is to design a robust IMPS that performs well for a broad set of performance conditions, e.g. different genres, styles, tempo e.t.c. Due to the diversity of the domain, this objective is rather difficult to achieve with rule-based approaches (Bresin, 2000).

Machine learning techniques provide particularly useful alternatives to rule-based systems. One powerful machine learning strategy is statistical modeling, i.e. to devise a probabilistic model with adjustable parameters. Then, optimal parameters are estimated by maximization of the likelihood on a representative dataset. In the context of interactive performance systems, model parameters are adapted to a particular performance situation or stylistic features of a specific composer/performer (Vercoe and Puckette, 1985; Thom, 2000; Raphael, 1999).

Usually training is accomplished off-line, i.e. there is an initial training phase when model parameters are adapted. Consequently, during the normal mode of operation parameters remain fixed. The fundamental problem with this conventional learning scenario is the difficulty in collecting a data set that represents all performance conditions one would be interested in. On the other hand, a model trained on a specialized data set might perform in a rather unexpected or unsatisfactory way on a novel domain.

Moreover using a large and inhomogeneous dataset may not necessarily result in "better" parameter estimates. In practice it is often possible to capture different stylistic aspects of an individual performer within a relatively simple model class. However, optimal parameters for different performers can be significantly different. Hence, a single set of parameters optimized for the entire dataset may lead to unsatisfactory performance. Additionally, for an individual performer, optimal parameters can change among different performances or even "drift away" during a particular performance situation. Therefore it is desirable to have a built in online adaptation schema that updates parameters during normal mode of operation.

2 Bayesian Parameter Estimation

In this section we introduce the key concepts of Bayesian parameter estimation on a probabilistic tempo tracking model. A tempo tracker can be considered as the backbone of any IMPS so robustness is of primary importance.

2.1 A generative model for tempo fluctuations

Consider the following recursion

$$\begin{pmatrix} \omega_i \\ a_i \end{pmatrix} = A(\theta) \begin{pmatrix} \omega_{i-1} \\ a_{i-1} \end{pmatrix} \tag{1}$$

where

$$A(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

is a rotation matrix that, when premultiplied, rotates a vector by θ degrees counterclockwise. Consequently, all points $x_i = (\omega_i, a_i)$ are located on a circle. Hence, the state variable ω_i, when viewed as a function of i, is a perfect sinusoidal function. The phase and amplitude of ω_i is determined by the initial conditions (ω_0, a_0). The frequency is determined by θ.

We can use ω_i to generate a regular beat with fluctuating tempo as $t_i = t_{i-1} + 2^{\omega_i}$, where t_i is the time when the i'th onset occurs.

Note that the above model is an entirely deterministic model for tempo fluctuations. In reality, we expect some random deviations so we introduce noise terms

$$\begin{pmatrix} \omega_i \\ a_i \end{pmatrix} = A(\theta_i) \begin{pmatrix} \omega_{i-1} \\ a_{i-1} \end{pmatrix} + \xi_i \tag{2}$$

$$t_i = t_{i-1} + 2^{\omega_i} + \epsilon_i \tag{3}$$

where ξ_i and ϵ_i are zero mean normal random variables with covariance matrices Q and R respectively. We will denote the multivariate gaussian distribution with mean μ and covariance matrix Σ with $\mathcal{N}(\mu, \Sigma)$. Moreover, if θ is assumed to vary, we will denote it by θ_i. In this example we assume that $R = 0$, i.e. ω_i is directly observed.

Given θ, Equation 2 defines implicitly a probability distribution $p(\boldsymbol{\omega}|\theta)$ over possible tempo trajectories. Moreover due to Gaussian noise and linear state transition assumptions, the distribution $p(\boldsymbol{\omega}|\theta)$ is also (a big) Gaussian.

In Figure 1, we plot two $\boldsymbol{\omega}$ sequences sampled from the model in Eq. 2. The sequences are drawn from a constant-θ and from a varying-θ model respectively. The constant-θ model has $\theta = 0.5$. In the varying-θ model, θ is interpolated linearly from 0.0 to 0.8. As expected, the samples have different characteristics. The constant-θ sequence $\boldsymbol{\omega}^{\text{const}}$ has roughly the same period throughout whereas the varying-θ sequence $\boldsymbol{\omega}^{\text{vary}}$ is "chirp-like", i.e. its frequency is increasing with i.

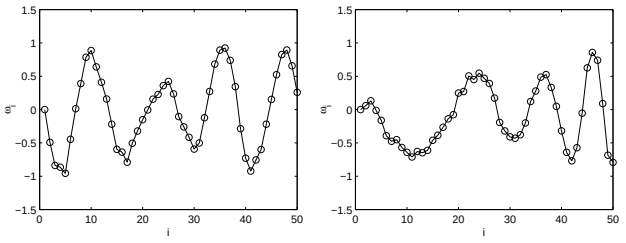

Figure 1: Typical "tempo curves" $\boldsymbol{\omega}$ sampled from Eq.2 with $Q = 0.1^2 I$. Left, constant $\theta = 0.5$, Right, θ_i is interpolated linearly from 0.0 to 0.8 in 50 steps.

This model is a constrained version of the Kalman filter introduced in (Cemgil et al., 2001). In the current model, the beat is not explicitely modeled and the state transition matrix A is constrained to be a rotation matrix with only one parameter. The last constraint is imposed to simplify the discussion and will be removed later.

2.2 Learning

Learning is the reverse problem of generating samples from a given model: we are given sequences and wish to estimate model parameters. In the following we wish to estimate a constant-θ model (we assume that Q is known). The Bayesian formulation of the problem is

$$p(\theta|\boldsymbol{\omega}) = \frac{p(\boldsymbol{\omega}|\theta)p(\theta)}{p(\boldsymbol{\omega})} \quad (4)$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}} \quad (5)$$

The prior term $p(\theta)$ reflects our knowledge about the parameter θ before we observe any data. In this example we take the prior $p(\theta)$ as uniform on $[0, 1]$. The likelihood term $p(\boldsymbol{\omega}|\theta)$ is a measure of how well a given θ *predicts* the data. Since in this toy example θ is just a scalar, we can plot the likelihood by evaluating it at several points on $[0, 1]$. Note that each different θ corresponds to a different Kalman filter. The likelihood at each θ is computed by running standard Kalman filtering recursion.

The resulting likelihood functions for both sequences are plotted in Figure 2. Note that the likelihood function is **not** a probability distribution of θ since it is not normalized. The required normalization constant, the evidence, is given $p(\boldsymbol{\omega}) = \int d\theta p(\boldsymbol{\omega}|\theta)p(\theta)$. The evidence plays a key role in Bayesian inference: it gives the likelihood that the model has generated the observed data by summing (integrating) individual parameter likelihoods over all possible parameter settings. The likelihood $p(\boldsymbol{\omega}|\theta)$ answers the question "what is the likelihood that the particular θ has generated the data (given the model)" whereas the evidence answers the "global" question "what is the likelihood that the data comes from a constant θ model". In this example the log-evidence is 21.58 and -4.78 respectively: It is about 11 orders of magnitude less likely that $\boldsymbol{\omega}^{\text{vary}}$ comes from a constant θ model.

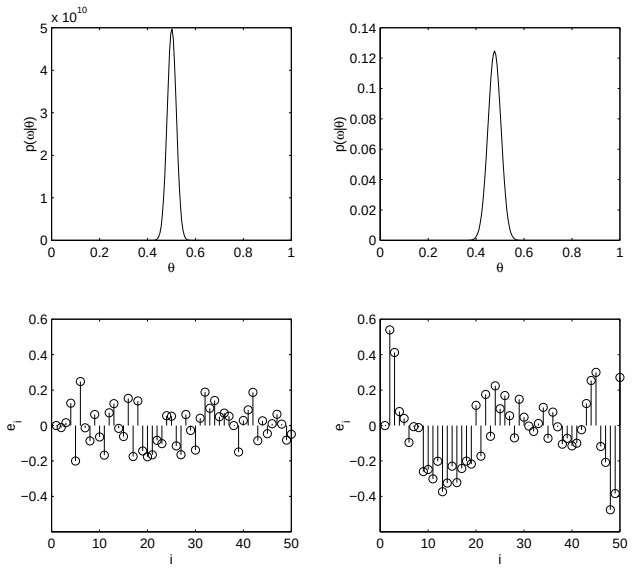

Figure 2: The likelihood functions (left-up) $p(\boldsymbol{\omega}^{\text{const}}|\theta)$ and (right-up) $p(\boldsymbol{\omega}^{\text{vary}}|\theta)$. Since the prior is flat, the posterior is proportional to the likelihood upto a normalization constant. Under each likelihood, the error signal $e_i = \omega_i - \omega_i^{\text{pred}}$ is shown for a Kalman filter with $\theta = 0.5$. The error signal for $\boldsymbol{\omega}^{\text{vary}}$ has higher magnitude and exhibits correlations that indicate the fact that the filter is unable to capture the structure in the signal.

The posterior distribution $p(\theta|\boldsymbol{\omega})$ reflects our entire knowledge about the parameter θ after we observe the data. In this respect the full Bayesian parameter estimation is the general case of maximum likelihood (ML) or maximum a-

posteriori (MAP) estimation, where one is interested into just a single parameter that maximizes $p(\boldsymbol{\omega}|\theta)$ or $p(\boldsymbol{\omega}|\theta)p(\theta)$ respectively. In other words, ML and MAP estimation can be viewed as ways of summarizing or approximating the underlying posterior distribution $p(\theta|\boldsymbol{\omega})$ by a point estimate.

However, the computation of the exact posterior distribution is usually intractable and one has to reside to approximation techniques. One such approximation method is variational approximation. In the context of the current example, we approximate the exact parameter posterior $p(\theta|\boldsymbol{\omega})$ by a Gaussian $\mathcal{N}(\mu_\theta, \Sigma_\theta)$ and estimate both the mean μ_θ parameter *and* the variance Σ_θ. As can be seen in Figure 2, a Gaussian approximation would be quite reasonable.

2.3 Variational Expectation Maximization

The well known Expectation Maximization (EM) algorithm for ML parameter estimation includes two steps that are iterated. In the E step the sufficient statistics (e.g. the sample mean and covariance) of the unobserved variables is estimated by fixing the parameters. Consequently, in the M step, the estimated statistics are fixed and the maximum likely parameter is computed. See Bishop (1995) for an introduction to EM.

The variational-EM (VEM) can be considered as an extension to EM where both E and M steps are "symmetric": In the VE step the sufficient statistics (e.g. the sample mean and covariance) of the unobserved variables is estimated by fixing the parameters. Consequently, in the VM step, the estimated statistics are fixed and the sufficient statistics of parameters is computed. Luckily, it turns out that the resulting variational algorithms have very little additional computation cost compared to ML version (Ghahramani and Beal, 2000).

3 Bayesian Online Adaptation

The example in the previous section demonstrated that if the signal characteristics are changing or parameters are not well tuned (consider the fact that the posterior in Fig. 2 is quite peaked) the predictions can be quite bad.

In this section we introduce an online learning mechanism to adapt model parameters. See Figure 3 for a sematic description of Bayesian online learning. The online formulation of variational Bayesian learning is simple: the parameter distribution is updated each time new data arrives. In other words, the previous parameter posterior acts as the prior of the next step. The parameter distribution is improved based on recent data by variational EM. Since VEM is guaranteed to improve the estimate at each step, the new parameter distribution is calculated as long as computational resources permit.

One additional advantage of keeping a distribution over the parameters is that the adaptation rate can be easily controlled: after each online update we can slightly increase the variance of the parameter distribution. In this way one prevents the parameter distribution shrinking to a point estimate and enables it to drift in time.

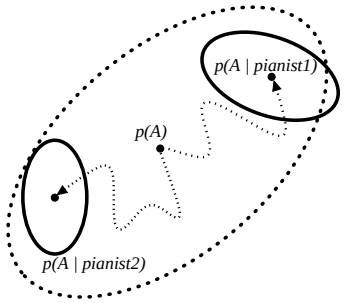

Figure 3: A Schematic representation of online learning. The "big" ellipse represents a distribution over plausible parameters. The mean of this distribution corresponds to some average parameter setting, that is potentially suboptimal for a particular pianist (or performance situation). In online adaptation, parameter distributions drift to the "smaller" ellipses and eventually capture the performer specific parameter distribution.

4 Model

The model introduced in section 2 assumed that the transition matrix is a rotation matrix and two dimensional. In general A can be an arbitrary $N \times N$ matrix. In (Cemgil et al., 2001) we have observed that higher order Kalman filters ($N \approx 10$) perform well. In this general case the hidden states of the Kalman filter correspond to the period and higher order acceleration terms of the tempo tracker. The parameters of a standard Kalman filter (in this particular case the transition matrix A) are fixed. We extend the original model such that filter parameters are also adapted online. The adaptive model is shown in Figure 4. Here, A_j denotes the transition matrix at step j. The rectangle denotes a sliding window of L steps. After each new observation, (1) the new hidden state distributions in the sliding window are calculated using the current parameter distribution, (2) the parameter distribution is updated using the recently obtained hidden states. Step 1 (Expectation) and 2 (Maximization) are iterated until a prediction has to be generated. We take a Gaussian distribution on each row of the state transition matrix A as in (Ghahramani and Beal, 2000). The prediction is calculated using the improved parameter estimate. When the new observation arrives, the window is shifted by one step and the whole procedure is repeated.

5 Results and Discussion

We compare the static model and the adaptive model by how well they predict the next beat in a given performance. The

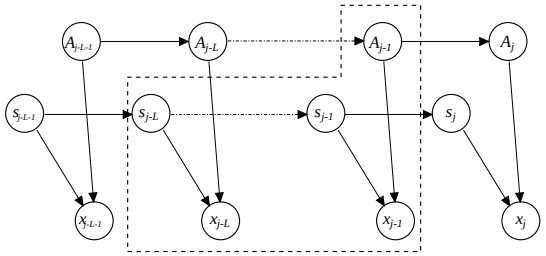

Figure 4: Graphical Model of the adaptive tempo tracker.

static filter uses parameters that are optimized for the entire dataset. The adaptive filter starts from a parameter distribution that has the same mean as the static distribution and a broad uncertainty (large variance). As a natural measure for prediction ability we use the log-likelihood of the next beat under this prediction, i.e. a quantity directly related to the prediction error. We found that a window length L of around 16 steps (4 bars) gives the best results. It has to be noted that the window size, as well as the initial parameter distribution are two factors that effect the rate of adaptation.

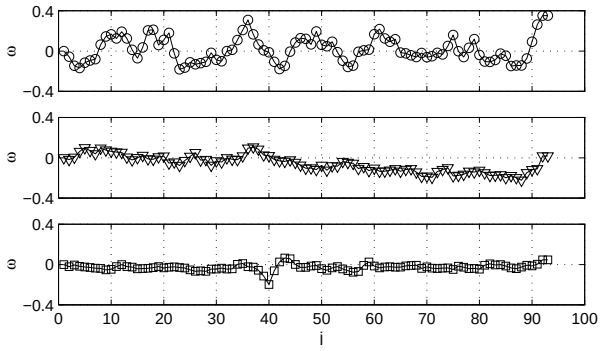

Figure 5: Examples of ω sequences from the Beatles data set. The sequences correspond (from above to below) to performances of a professional classical, amateur classical and professional jazz pianist. The performances have different characteristics. For example the classical pianist uses a lot more tempo fluctuation than the professional jazz pianist. The amateur "rushes", i.e. constantly accelerates.

For our simulations we have used 108 piano performances of Michelle by the Beatles. This dataset is introduced in (Cemgil et al., 2001). See Figure 5 for a few examples of estimated ω sequences. In Figure 6 we show the histogram of the likelihood differences of static and adaptive filters. On average, adaptation results in better predictions. For some performances the static filter is slightly better. Here, the adaptive filter merely learns some unstructured fluctuations. However, for the majority of examples the prediction accuracy improves, and sometimes quite significantly. For example the rightmost 3 performances (where the log-likelihood increases

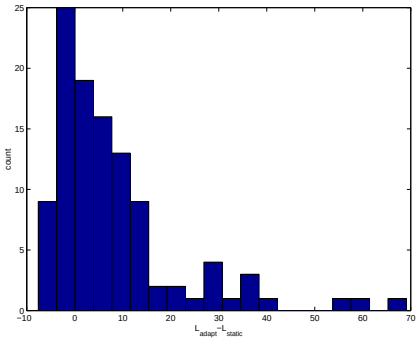

Figure 6: Histogram of the log-likelihood differences $\Delta\mathcal{L} = \mathcal{L}_{adapt} - \mathcal{L}_{static}$ for 108 performances of Michelle. A positive difference indicates that the adaptive model predicts better.

by more than 50) correspond to the same subject who uses consistently a lot of tempo variation. Hence, her "personal" optimal parameters are significantly different than other performers.

These result suggests that online adaptation has the potential to capture structure in expressive performances. Moreover, variational Bayesian techniques seem to be an efficient and stable way to accomplish this goal in realtime.

References

Bishop, C. 1995. *Neural Networks for Pattern Recognation*. Oxford University Press.

Bresin, R. 2000. *Virtual Virtuosity - Studies in automatic music performance*. PhD thesis, Speech Music and Hearing, Stockholm.

Cemgil, A. T., Kappen, H., Desain, P., and Honing, H. 2001. "On tempo tracking: Tempogram representation and kalman filtering". *Journal of New Music Research*.

Ghahramani, Z. and Beal, M. 2000. "Propagation algorithms for variational bayesian learning". In *Neural Information Processing Systems 13*. `www.gatsby.ucl.ac.uk/~zoubin/papers.html`.

Raphael, C. 1999. "A probabilistic expert system for automatic musical accompaniment". *Journal of Computational and Graphical Statistics*, Accepted for Publication.

Rowe, R. 1993. *Interactive Music Systems : Machine Listening and Composing*. MIT Press.

Thom, B. 2000. "Unsupervised learning and interactive jazz/blues improvisation". In *Proceedings of the AAAI2000*. AAAI Press.

Vercoe, B and Puckette, M. 1985. "The synthetic rehearsal: Training the synthetic performer". In *Proceedings of ICMC*, San Francisco. International Computer Music Association, pages 275–278.