PARTICLE FILTERS FOR EFFICIENT METER TRACKING WITH DYNAMIC BAYESIAN NETWORKS

Ajay Srinivasamurthy[*]
ajays.murthy@upf.edu

Andre Holzapfel[†]
andre@rhythmos.org

Ali Taylan Cemgil[†]
taylan.cemgil@boun.edu.tr

Xavier Serra[*]
xavier.serra@upf.edu

[*]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[†]Dept. of Computer Engineering, Boğaziçi University, Istanbul, Turkey

ABSTRACT

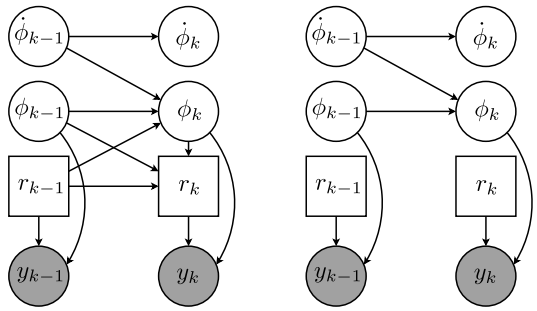
Recent approaches in meter tracking have successfully applied Bayesian models. While the proposed models can be adapted to different musical styles, the applicability of these flexible methods so far is limited because the application of exact inference is computationally demanding. More efficient approximate inference algorithms using particle filters (PF) can be developed to overcome this limitation. In this paper, we assume that the type of meter of a piece is known, and use this knowledge to simplify an existing Bayesian model with the goal of incorporating a more diverse observation model. We then propose Particle Filter based inference schemes for both the original model and the simplification. We compare the results obtained from exact and approximate inference in terms of meter tracking accuracy as well as in terms of computational demands. Evaluations are performed using corpora of Carnatic music from India and a collection of Ballroom dances. We document that the approximate methods perform similar to exact inference, at a lower computational cost. Furthermore, we show that the inference schemes remain accurate for long and full length recordings in Carnatic music.

1. INTRODUCTION

Rhythm analysis of musical audio signals plays an important role in Music Information Retrieval (MIR) research. Many of the works in MIR related to rhythm attempt to establish a relation between the audio signal and the underlying musical meter. For instance, in the task of beat tracking, the goal is to obtain an alignment of the metrical level referred to as the *tactus* [15] to an audio signal, see [8] for a list of references to recent beat tracking algorithms. Tracking meter at a higher metrical level is a task pursued under the title of downbeat detection. Approaches were presented that either attempt to identify the downbeat separately from the tactus [7], or that pursue beat tracking and downbeat detection as a combined task [11, 17]. The combined task of beat and downbeat detection is what we refer to as meter tracking, since it aims at aligning several levels of a known meter to an audio recording of a music performance.

Many applications can profit from accurate meter or beat tracking. Some synchronization tasks, such as the one presented in [6], tracking the beat is sufficient. However, other applications, such as musical structure analysis [16] can profit from a more detailed understanding of the temporal structure of a performance. Approaches that can achieve such an analysis for a wider variety of music usually incorporate machine learning strategies to adapt to new styles. For instance, Böck et al. [1] presented a method for beat tracking in various styles that achieves high accuracy using recurrent neural networks that were adapted to the individual styles. The task of downbeat tracking was addressed in [4] using a set of deep belief networks trained on various features, and the regularity of the outputs was enforced by incorporating a simple hidden Markov model (HMM). The task of meter tracking was combined with the determination of the type of meter in [9], using a Dynamic Bayesian Network (DBN) similar to the one applied in [1].

A significant shortcoming of the mentioned tracking approaches is that their flexibility in terms of musical style comes at an increased computational cost, either in terms of time spent for the training of networks [1, 4], or in terms of long inference times [9]. In the present paper, we approach faster inference in a DBN in two ways. Firstly, we propose a change to the model structure as presented in [9, 14] that enables faster inference by simplifying the independence assumptions between the variables of the model. The proposed simplification also addresses one of the main limiting factors in most of the approaches so far: a simplistic observation model that cannot effectively handle diversity in rhythmic patterns. Secondly, one reason for long inference times of the model proposed in [9] is the utilization of exact inference in an HMM, which discretizes the hidden variables of the state space to compute the most likely path in the exact posterior distribution using the Viterbi algorithm. Here, we avoid the discretization of the state space by approximating the posterior using particle filter methods [3]. The biggest challenge in applying such approximate methods to meter tracking is the multi-modality of the underlying posterior distribution [22] due to the ambiguity inherent to musical meter. Recently, methods were proposed that overcome these challenges [14]. We outline the existing [9, 14] and the proposed simplified model, and compare the performance of exact and approximate inference schemes for both the models, in terms of meter track-



(a) model-A (b) model-B

Figure 1: The DBNs used in this paper: circles and squares denote continuous and discrete variables, respectively. Gray nodes and white nodes represent observed and latent variables, respectively. Model-A is from [14] and model-B is the proposed simplification.

ing accuracy and computational demands.

Carnatic music, the art music tradition from South India is a representative case to study in this context. Meter in Carnatic music is defined by the *tāḷa*, which are time cycles with three metrical levels: the *sama* (downbeat, the first pulse of the cycle), beat, and the subdivision level (a comprehensive account on Carnatic music is provided in [19]). In performances of Carnatic music, however, large degrees of freedom are taken by the musicians to conceal the underlying meter and to add metrical ambiguity, for instance by changing the beat structure during a metrical cycle. This playful rhythmic character of Carnatic music leads to our hypothesis that meter tracking should be able to profit from a diverse observation model. Most of the rhythmic structures, melodic phrases, and strutural elements are tightly associated with the cycles of the tāḷa [20] and hence tracking the sama (downbeat) is an important MIR task in Carnatic music, which is the main focus of this paper. We will also evaluate if meter tracking in Carnatic music can profit from including a richer observation model that can incorporate information from multiple patterns.

In order to further illustrate the ability of the approach to generalize, it will be additionally evaluated on a corpus of Ballroom dances [5]. Furthermore, reproducibility will be ensured by providing free access for research purposes to all code repositories and datasets [1] . We begin by describing the models and inference schemes that we use for meter tracking.

2. MODEL STRUCTURE

We compare two different Bayesian models for the task of meter tracking. The first model (model-A), depicted in Figure 1a, is identical to the model used in [9, 14] and was initially proposed in [24]. We propose and discuss a simplification to model-A for the task of meter tracking, shown as model-B in Figure 1b. Model-B uses a diverse observation model and can be applied if the type of meter is known in advance. It is to be noted that model-A can also be used for inferring the type of meter, though we apply it in this paper only for meter tracking.

[1] Please see the companion webpage for more details: `http://compmusic.upf.edu/ismir-2015-pf`

In a DBN, an observed sequence of features derived from an audio signal $\mathbf{y}_{1:K} = \{\mathbf{y}_1, \ldots, \mathbf{y}_K\}$ is generated by a sequence of hidden (unknown) variables $\mathbf{x}_{1:K} = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$, where K is the length of the sequence (number of audio frames in an audio excerpt). The joint probability distribution of hidden and observed variables factorizes as,

$$P(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}) = P(\mathbf{x}_0) \cdot \prod_{k=1}^{K} P(\mathbf{x}_k|\mathbf{x}_{k-1})\, P(\mathbf{y}_k|\mathbf{x}_k) \quad (1)$$

where, $P(\mathbf{x}_0)$ is the initial state distribution, $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition model, and $P(\mathbf{y}_k|\mathbf{x}_k)$ is the observation model.

2.1 Hidden Variables

At each audio frame k, the hidden variables describe the state of a hypothetical bar pointer $\mathbf{x}_k = [\phi_k\ \dot{\phi}_k\ r_k]$, representing the bar position, instantaneous tempo and a rhythmic pattern indicator, respectively (see Figure 1 of [23] for an illustration).

- *Bar position*: The bar position $\phi \in [0, M)$, where M is the length of the bar (cycle). The maximum value of M depends on the longest bar (cycle) that is tracked. We set the length of a full note to 1600, and scale other bar (cycle) lengths accordingly.
- *Rhythmic pattern*: The rhythmic pattern variable $r \in \{1, \ldots, R\}$ is an indicator variable to select one of the R observation models corresponding to each bar (cycle) length rhythmic pattern learned from data. Each pattern has a bar length M and a number of beats B, which are assumed to be known in advance, i.e. the goal is the tracking of a known metrical structure.
- *Instantaneous tempo*: Instantaneous tempo $\dot{\phi}$ is the rate at which the bar position variable progresses through the cycle at each time frame, measured in bar positions per time frame. The range of the variable $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ depends on the length of the cycle M and the hop size ($\Delta = 0.02s$ used in this paper), and can be preset or learned from data. A tempo value of $\dot{\phi}_k$ corresponds to a bar (cycle) length of $(\Delta \cdot M/\dot{\phi}_k)$ seconds and $(60 \cdot {}^{B\cdot\dot{\phi}_k}/_{(M\cdot\Delta)})$ beats per minute.

The conditional dependence relations between the variables for both the models are shown in Figure 1.

2.2 Initial state distribution

We can use $P(\mathbf{x}_0)$ to incorporate prior information about the metrical structure of the music into the model. In this paper, we assume uniform priors on all variables, within the allowed ranges of tempo.

2.3 Model-A: Transition and Observation model

Due to the conditional dependence relations in Figure 1a, the transition model factorizes as,

$$P(\mathbf{x}_k|\mathbf{x}_{k-1}) = P(\phi_k|\phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) P(\dot{\phi}_k|\dot{\phi}_{k-1}) \times P(r_k|r_{k-1}, \phi_k, \phi_{k-1}) \quad (2)$$

Each of the terms in Eqn (2) are defined in Eqns (3)–(5).

$$P(\phi_k|\phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) = \mathbb{1}_{\phi} \quad (3)$$

where $\mathbb{1}_{\phi}$ is an indicator function that takes a value of one if $\phi_k = (\phi_{k-1} + \dot{\phi}_{k-1}) \bmod(M(r_k))$ and zero otherwise (in our case, $M(r_k) = M$), meaning that the bar position advances at the rate of the instantaneous tempo variable, and folds back when it crosses the maximum value that is defined by the length M of the metrical cycle.

$$P(\dot{\phi}_k|\dot{\phi}_{k-1}) \propto \mathcal{N}(\dot{\phi}_{k-1}, \sigma^2_{\dot{\phi}}) \times \mathbb{1}_{\dot{\phi}} \tag{4}$$

where $\mathbb{1}_{\dot{\phi}}$ is an indicator function that equals one if $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ and zero otherwise. $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ.

$$P(r_k|r_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbf{A}(r_{k-1}, r_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_r & \text{else} \end{cases} \tag{5}$$

where, $\mathbf{A}(i, j)$ is the time-homogeneous transition probability from r_i to r_j, and $\mathbb{1}_r$ is an indicator function that equals one when $r_k = r_{k-1}$ and zero otherwise. Since the rhythmic patterns are one bar (cycle) in length, pattern transitions are allowed only at the end of the bar (cycle). The pattern transition probabilities are learned from data.

The observation model is identical to the one used in [14], and depends only on the bar position and rhythmic pattern variables. We use a two dimensional spectral flux feature in two frequency bands (Low: $\leq$ 250 Hz, High: $>$ 250 Hz). Using beat and downbeat annotated training data, a k-means clustering algorithm clusters and assigns each bar of the dataset (represented by a point in a 128-dimensional space) to one of the R rhythmic patterns. We then discretize the bar into 64$^{\text{th}}$ note cells (corresponding to 25 bar positions with $M_{\text{max}} = 1600$), collect all the features within the cell for each pattern, and compute the maximum likelihood estimates of the parameters of a two component Gaussian Mixture Model (GMM). The observation probability hence is computed as,

$$P(\mathbf{y}|\mathbf{x}) = P(\mathbf{y}|\phi, r) = \sum_{i=1}^{2} w_{\phi,r,i}\, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\phi,r,i}, \boldsymbol{\Sigma}_{\phi,r,i}) \tag{6}$$

where, $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution and for the mixture component i, $w_{\phi,r,i}$, $\boldsymbol{\mu}_{\phi,r,i}$ and $\boldsymbol{\Sigma}_{\phi,r,i}$ are the component weight, mean (2-dimensional) and the covariance matrix (2×2), respectively.

2.4 Model-B: Transition and Observation model

We propose a simpler model-B (Figure 1b) that uses a diverse mixture observation model incorporating observations from multiple rhythmic patterns. Since all the rhythmic patterns belong to the same type of meter (tāḷa), we can simplify model-A to track only the ϕ and $\dot{\phi}$ variables while using an observation model that computes the likelihood of an observation by marginalizing over all the patterns. The motivation for this simplification is two-fold: the inference is simplified, and we can increase the influence of diverse patterns that occur throughout a metrical cycle in the inference.

For model-B, we first define $\mathbf{x}_k = [\boldsymbol{\alpha}_k, r_k]$, where $\boldsymbol{\alpha}_k = [\phi_k, \dot{\phi}_k]$. Based on the conditional dependence relations in Figure 1b, the transition model now is,

$$P(\mathbf{x}_k|\mathbf{x}_{k-1}) = P(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k-1}) = P(\phi_k|\phi_{k-1}, \dot{\phi}_{k-1}) P(\dot{\phi}_k|\dot{\phi}_{k-1}) \tag{7}$$

Eqns. (3) and (4) remain identical apart from the removal of the dependence on r_{k-1} in Eqn (3). The observation model is a pre-computed mixture observation model computed from Eqn (6) by marginalizing over the patterns, assuming equal priors.

$$P(\mathbf{y}|\boldsymbol{\alpha}) \propto \sum_{j=1}^{R} P(\mathbf{y}|\phi, r = j) \tag{8}$$

3. INFERENCE METHODS

The goal of inference is to find a hidden variable sequence that maximizes the posterior probability of the hidden states given an observed sequence of features: a maximum *a posteriori* (MAP) sequence $\mathbf{x}^*_{1:K}$ that maximizes $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$. The inferred hidden variable sequence $\mathbf{x}^*_{1:K}$ can then be translated into a sequence downbeat (sama) instants ($\phi^*_k = 0$), beat instants ($\phi^*_k = i \cdot M/B$, $i = 1, \ldots, B$), and the local instantaneous tempo ($\dot{\phi}^*_k$). We describe two different inference schemes, an exact inference using an HMM in a discretized state space, and an approximate inference using particle filters using the continuous values of ϕ and $\dot{\phi}$.

3.1 Hidden Markov model (HMM)

By discretizing the continuous variables bar position and tempo, we can perform an exact inference using HMM. We use the discretization proposed in [14], by replacing the continuous variables ϕ and $\dot{\phi}$ by their discretized counterparts, $m \in \{1, 2, \ldots, \lceil M \rceil\}$ and $n \in \{n_{\text{min}}, n_{\text{min}} + 1, \cdots, n_{\text{max}}\}$, with the discrete tempo limits as $n_{\text{min}} = \lfloor \dot{\phi}_{\text{min}} \rfloor$ and $N = n_{\text{max}} = \lceil \dot{\phi}_{\text{max}} \rceil$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceil and floor operations, respectively. Eqns (2), (3) and (5) remain valid. We define the tempo transition probability within the allowed tempo range as,

$$P(n_k|n_{k-1}) = \begin{cases} 1 - p_n & \text{if } n_k = n_{k-1} \\ \frac{p_n}{2} & \text{if } n_k = n_{k-1} \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where p_n is the probability of tempo change. We use Viterbi algorithm [18] to obtain a MAP sequence of states with the HMM. We refer to the HMMs for inference from model-A and model-B as HMM_{a} and HMM_{b}, respectively.

The drawback of this approach is that the discretization has to be on a very fine grid in order to guarantee good performance, which leads to a prohibitively large state space and, as a consequence, to a computationally demanding inference. The size of the state space is $S = M \cdot N \cdot R$ and needs an $S \times S$ sized transition matrix. As an example, dividing a bar into $M = 1600$ position states, with $N = 15$ tempo states and $R = 4$ patterns, the size of the state space is $S = 96000$ states. The computational complexity of the Viterbi algorithm is $O(K \cdot |S|^2)$. Even though the state transition matrix is sparse due to lesser number of allowed transitions leading to a complexity of $O(K \cdot M \cdot R)$, the inference with HMM can become computationally prohibitive and does not scale well with increasing number of states. This problem can be overcome, for instance, by using approximate inference methods such as particle filters.

3.2 Particle Filter (PF)

Particle filters (or Sequential Monte Carlo methods) are a class of approximate inference algorithms to estimate the

posterior density of a state space. They overcome two main problems of the HMM: discretization of the state space and the quadratic scaling up of the size of state space with more number of variables. In addition, they can incorporate long term relationships between hidden variables.

The exact computation of the posterior $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ is often intractable, but it can be evaluated pointwise. In particle filters, the posterior is approximated using a weighted set of points (known as particles) in the state space as,

$$P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K}) \approx \sum_{i=1}^{N_p} w_K^{(i)} \delta(\mathbf{x}_{1:K} - \mathbf{x}_{1:K}^{(i)}) \quad (10)$$

Here, $\{\mathbf{x}_{1:K}^{(i)}\}$ is a set of points (particles) with associated weights $\{w_K^{(i)}\}$, $i = 1, \ldots, N_p$, and $\mathbf{x}_{1:K}$ is the set of all state trajectories until frame K, while $\delta(x)$ is the Dirac delta function, $\delta(x) = 1$ if $x = 0$ and 0 otherwise. N_p is the number of particles.

To approximate the posterior pointwise, we need a suitable method to draw samples $\mathbf{x}_k^{(i)}$ and compute appropriate weights $w_k^{(i)}$ recursively at each time step. A simple approach is Sequential Importance Sampling (SIS) [3], where we sample from a *proposal* distribution $Q(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ that has the same support and is as similar to the true (target) distribution $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ as possible. To account for the fact that we sampled from a proposal and not the target, we attach an importance weight $w_K^{(i)}$ to each particle, computed as,

$$w_K^{(i)} = \frac{P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})}{Q(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})} \quad (11)$$

With a suitable proposal density, these weights can be computed recursively as,

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{P(\mathbf{y}_k|\mathbf{x}_k^{(i)}) P(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{Q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \quad (12)$$

Following [14], we choose to sample from the transition probability $Q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k) = P(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$, which reduces Eqn (12) to

$$w_k^{(i)} \propto w_{k-1}^{(i)} P(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \quad (13)$$

The SIS algorithm derives samples by first sampling from proposal, in this case the transition probability and then computes weights according to Eqn (13). Once we determine the particle trajectories $\{\mathbf{x}_{1:K}^{(i)}\}$, we then select the trajectory $\mathbf{x}_{1:K}^{(i^*)}$ with the highest weight $w_K^{(i^*)}$ as the MAP state sequence.

Many extensions have been proposed to the basic SIS filter (see [3] for a comprehensive overview) to address several problems with it. We briefly mention some of the relevant extensions, emphasizing their key aspects. A more detailed description of the algorithms has been presented in [14]. The most challenging problem in particle filtering is the degeneracy problem, where within a short time, most of the particles have a weight close to zero, representing unlikely regions of state space. This is contrary to the ideal case when we want the proposal to match well with the target distribution leading to a uniform weight distribution with low variance. To reduce the variance of the particle weights, resampling steps are necessary, which replaces low weight particles with higher weight particles by selecting particles with a probability proportional to their weights. Several resampling methods have been proposed, but we use systematic resampling in this paper as recommended in [3]. With resampling as the essential difference, the SIS filter with resampling is called as Sequential Importance Sampling/Resampling (SISR) filter.

In meter tracking, due to metrical ambiguities, the posterior distribution $P(\mathbf{x}_k|\mathbf{y}_{1:k})$ is highly multimodal. Resampling tends to lead to a concentration of particles in one mode of the posterior, while the remaining modes are not covered. One way to alleviate this problem is to compress the weights $\mathbf{w}_k = w_k^{(i)}$, $i = 1, \ldots, N_p$ by a monotonically increasing function to increase the weights of particles in low probability regions so that they can survive resampling. After resampling, the weights have to be uncompressed to give a valid probability distribution. This can be formulated as an Auxiliary Particle Filter (APF) [10]. Further, a system that is capable of handling metrical ambiguities must maintain this multimodality and be able to track several hypotheses together, which SISR and APF cannot do explicitly. A system called the Mixture Particle Filter (MPF) was proposed to track multiple hypotheses in [22], and was adapted to meter inference in [14].

In an MPF, each particle is assigned to a cluster that (ideally) represents a mode of the posterior. During resampling, the particles of a cluster interact only with particles of the same cluster. Resampling is done independently in each cluster, while maintaining the probability distribution intact. This way, all the modes of the posterior can be tracked through the whole audio piece, and the best hypothesis can be chosen at the end. We use an identical clustering scheme using a cyclic distance measure as described in [14] to track several different possible metrical positions at a given time. In the MPF, after an initial cluster assignment, we perform a re-clustering before every resampling step, merging or splitting clusters based on the average distance between cluster centroids. The clustering, merging and splitting of clusters is necessary to control the number of clusters, which ideally represents the number of modes in the posterior. The mixture particle filter can be combined with the Auxiliary resampling to give the Auxiliary Mixture Particle Filter (AMPF). As recommended in [14], we resample at a fixed interval T_s. It was shown in [14] that AMPF can be effectively used for the task of meter inference and tracking.

With model-A, we setup an AMPF (AMPF_a) to compute the pointwise estimates of the posterior of $\mathbf{x}_{1:K}$, represented by $\left\{w_{\mathbf{x},K}^{(i)}, \mathbf{x}_{1:K}^{(i)}, i = 1, \cdots, N_p\right\}$, where N_p is the number of particles and $w_{\mathbf{x},K}^{(i)}$ are the weights corresponding to the particle trajectories $\mathbf{x}_{1:K}^{(i)}$. The weights are updated as in Eqn (13), using the observation model in Eqn (6). This particle filter is identical to the AMPF described in [14], however, in this paper it is evaluated for the first time assuming several patterns with transitions allowed.

For the simplified model-B, we setup AMPF_b similarly for $\boldsymbol{\alpha}_{1:K}$, represented by $\left\{w_{\boldsymbol{\alpha},K}^{(i)}, \boldsymbol{\alpha}_{1:K}^{(i)}, i = 1, \cdots, N_p\right\}$, where $w_{\boldsymbol{\alpha},K}^{(i)}$ are the weights corresponding to the particle trajectories $\boldsymbol{\alpha}_{1:K}^{(i)}$. Similar to Eqn (13), the weight updates

Algorithm 1 Outline of the AMPF$_b$ algorithm

for i = 1 to N_p **do**
  Sample $(\boldsymbol{\alpha}_0^{(i)}) \sim P(\phi_0)P(\dot{\phi}_0)$, set $w_{\boldsymbol{\alpha},0}^{(i)} = 1/N_p$
Cluster $\{\phi_0^{(i)}\}$ and obtain cluster assignments $\{c_0^{(i)}\}$
for k = 1 to K **do**
  for i = 1 to N_p **do**
    Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)}|\phi_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
    $\tilde{w}_{\boldsymbol{\alpha},k}^{(i)} = w_{\boldsymbol{\alpha},k}^{(i)} \times \sum_{j=1}^{R} P(\mathbf{y}_k|\phi_k^{(i)}, r = j)$
  for i = 1 to N_p **do** ▷ Normalize weights
    $w_{\boldsymbol{\alpha},k}^{(i)} = \frac{\tilde{w}_{\boldsymbol{\alpha},k}^{(i)}}{\sum_{i=1}^{N_p} \tilde{w}_{\boldsymbol{\alpha},k}^{(i)}}$
  if mod $(k, T_s) = 0$ **then**
    Recluster and Resample $\{\boldsymbol{\alpha}_k, w_{\boldsymbol{\alpha},k}\}$ and obtain $\{\hat{\boldsymbol{\alpha}}_k, \hat{w}_{\boldsymbol{\alpha},k}\}$, update $\{c_k^{(i)}\}$
    for i = 1 to N_p **do**
      Set $\boldsymbol{\alpha}_k^{(i)} = \hat{\boldsymbol{\alpha}}_k^{(i)}$, $w_{\boldsymbol{\alpha},k} = \hat{w}_{\boldsymbol{\alpha},k}$
  Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)}|\dot{\phi}_{k-1}^{(i)})$

for AMPF$_b$ are,

$$w_{\boldsymbol{\alpha},k}^{(i)} \propto w_{\boldsymbol{\alpha},k-1}^{(i)} P(\mathbf{y}_k|\boldsymbol{\alpha}_k^{(i)}) \tag{14}$$

where $P(\mathbf{y}_k|\boldsymbol{\alpha}_k^{(i)})$ is computed as in Eqn (8) by marginalizing $P(\mathbf{y}_k|\mathbf{x}_k^{(i)})$ over $r_k^{(i)}$. The AMPF$_b$ enables therefore to incorporate the full expressivity of the observed patterns into the inference. An outline of AMPF$_b$ is provided in Algorithm 1.

The complexity of the PF schemes scale linearly with N_p irrespective of the size of state space, leading to an efficient inference in large state spaces. Further, compared to the HMM using Viterbi decoding that has a space complexity of $O(K \cdot |S|)$, the PF needs to store just N_p state trajectories and weights, significantly reducing the memory requirements. An additional advantage is that the number of particles can be chosen based on the computational power we can afford, and we can make the state space larger with no or only a marginal increase in the computational requirements. Since the observation likehood can be precomputed, inference with model-B requires much lower computational resources, with only a marginal increase in cost during inference with increase in number of patterns.

4. EXPERIMENTS

The experiments aim to compare the performance of the particle filter and the HMM inference schemes for meter tracking with both model-A and model-B. Further, we wish to see if using a larger number of patterns per rhythm class (tāḷa) improves meter tracking performance. Meter tracking is done for each type of meter (tāḷa) separately, in a two fold cross validation experiment.

4.1 Music Corpora

The primary dataset we evaluate on is the Carnatic music dataset (CMD) used in [9]. It includes 118 two minute long excerpts spanning four commonly used tāḷas as shown in Table 1, with a total duration of 236 minutes and over 5500 sama instances. To test if the results extend to full pieces, we use the super set of CMD consisting of longer and full length pieces (called CMD$_f$) as used in [21]. CMD$_f$ comprises about 16.6 hours of audio with over 22600 sama instances. For comparability, we also present results on the Ballroom dataset [5], using the annotations from [12].

Tāḷa	M	B	#Excerpts CMD	#Pieces CMD$_f$
Ādi (8/8)	1600	8	30 (60)	50 (252.8)
Rūpaka (3/4)	1200	3	30 (60)	50 (267.4)
Miśra chāpu (7/8)	1400	7	30 (60)	48 (342.1)
Khaṇḍa chāpu (5/8)	1000	5	28 (56)	28 (134.6)

Table 1: The Carnatic music datasets, showing the cycle length M used in the paper and the number of beats B for each tāḷa. The analogous time signature is also shown. CMD is a subset of CMD$_f$, with two minute excerpts from full pieces. The number of pieces/excerpts in both datasets is also shown, the numbers in parentheses indicate the total duration of audio in minutes.

4.2 Parameter Selection and Learning

The tempo ranges were manually set for Carnatic music as $\dot{\phi} \in [4, 15]$ (cycle lengths between 1.33 s and 8 s) and $\dot{\phi} \in [6, 32]$ (bar lengths between 0.75 s to 5.3 s) for the Ballroom dataset. With $M_{\max} = 1600$ (corresponds to ādi tāḷa with 8 beats/cycle), the length of cycle M and the number of beats B for each tāḷa is shown in Table 1. For Ballroom dataset, we used $M = 1600$ and $M = 1200$ for tracking time signatures 4/4 and 3/4, respectively. For the HMM, we use $p_n = 0.02$ as in [12], and for the AMPF, we use $\sigma_{\dot{\phi}} = 10^{-4} \cdot M$. We explore the performance with $R = \{1, 2, 4\}$, with the number of particles set to $N_p = 1500 \cdot R$. The other AMPF parameters are identical to the values used in [14].

4.3 Evaluation Measures

A variety of measures for evaluating beat and downbeat tracking performance are available (see [2] for a detailed overview and descriptions of the metrics listed below [2]). We chose two metrics that are characterized by a set of diverse properties and are widely used in beat tracking evaluation. We describe it for beats, but the definitions extend to downbeats/samas as well, with the same tolerances. We use the prefix 's-' and 'b-' to distinguish between the performance measures of sama and beat tracking, respectively.

Fmeas (F-measure): The F-measure (a number between 0 and 1) is computed from correctly detected beats within a window of ± 70 ms as the harmonic mean of the precision (the ratio between the number of correctly detected beats and all detected beats) and recall (the ratio between the number of correctly detected beats and the total annotated beats).

AMLt (Allowed Metrical Levels with no continuity required): In the AMLt measure (a number between 0 and 1), beat sequences are considered as correct if the beats occur on the off-beat, or are double or half of the annotated tempo, allowing for metrical ambiguities. The value of this

[2] We used the code available at `http://code.soundsoftware.ac.uk/projects/beat-evaluation/` with default settings

	Sama tracking						Beat tracking					
Measure	s-Fmeas			s-AMLt			b-Fmeas			b-AMLt		
R	1	2	4	1	2	4	1	2	4	1	2	4
HMM_a	0.733	0.736	0.713	0.837	0.837	0.804	0.85	0.847	0.850	0.868	0.874	0.852
$AMPF_a$	0.708	0.697	0.704	0.827	0.809	0.822	0.846	0.833	0.843	0.872	0.874	0.862
HMM_b	0.726	0.735	0.736	0.830	0.862	0.867	0.844	0.849	0.837	0.864	0.893	0.900
$AMPF_b$	0.690	0.712	0.735	0.832	0.842	0.853	0.833	0.838	0.846	0.869	0.888	0.890
Klapuri [11]	0.175			0.181			0.657			0.650		

Table 2: Meter tracking performance on CMD. In addition, the performance of meter tracking with the algorithm proposed in [11] is also shown for reference.

Dataset	CMD_f		Ballroom	
Measure	s-Fmeas	b-Fmeas	s-Fmeas	b-Fmeas
HMM_a	0.727	0.834	0.806	0.929
$AMPF_b$	0.728	0.834	0.793	0.930

Table 3: F-measure for meter tracking on CMD_f and the Ballroom dataset, with $R = 4$. Values in each column are not statistically significantly different.

measure is then the ratio between the number of correctly estimated beats divided by the number of annotated beats.

4.4 Results and Discussion

We report the average Fmeas and AMLt values for all excerpts over all the tāḷas for the HMM and AMPF schemes in Table 2. The results for AMPF are the mean values over three experiments. We conducted evaluations using several other measures as well without any qualitative change in results. Therefore, experimental results are documented using these two measures. We use a three-way ANOVA with tāḷa, inference scheme, and R as factors to assess statistically significant differences (at 5% significance levels).

In general, we see that the beat tracking performance is similar across all the inference schemes and values of R, with the b-Fmeas and b-AMLt values being comparable. This shows that adding a diverse observation model and additional patterns does not add a significant change, showing that handling pattern diversity is not needed for beat tracking.

For sama tracking, we see that the AMPFs show statistically equivalent performance to the HMMs. The simpler $AMPF_b$ performs as good or better than $AMPF_a$, with a lower computational complexity. Higher number of patterns ($R > 1$) do not show significant improvement in tracking performance, despite a richer observation model. This observation needs further exploration to verify if incorporating more patterns with the currently used features helps to improve sama tracking. Further, s-AMLt is significantly larger than s-Fmeas and shows that there is a potential for improvement in tracking the correct metrical level.

Though we report only consolidated set of results averaged over all the tāḷas, the tracking performance is significantly poorer for ādi tāḷa (e.g. s-Fmeas = 0.4, b-Fmeas = 0.632 with $AMPF_b$ and $R = 4$), with superior (and statistically equivalent) results with other three tāḷas (e.g. s-Fmeas = 0.849, b-Fmeas = 0.92 with $AMPF_b$ and $R = 4$). This is attributed to the long cycle durations and a large variety of patterns in ādi tāḷa, which shows a definite scope for improvement using higher number of patterns and better observation models.

We extend the evaluation and report the performance of HMM_a and the proposed $AMPF_b$ on CMD_f and Ballroom datasets (in an identical setting, assuming that the meter type is known) in Table 3. We see that the observations from CMD extend to these datasets too. We further see a similar performance between CMD and CMD_f, that shows that the AMPF generalizes to longer and full length pieces.

One of the main advantages of model-B over model-A is the lower computational cost. For meter tracking under the conditions described, all the inference schemes have faster than real time execution. Inference in model-B is faster than that in model-A: model-B speeds up inference by a factor of about 5 for HMM and 2.5 for AMPF (for $R = 4$ and ādi tāḷa). Even in the smaller state space with model-B, HMM_b has a higher memory requirement than $AMPF_b$, which shows the utility of PF inference schemes.

5. CONCLUSIONS

For the task of meter tracking, we presented a simplified Bayesian model that incorporates a richer observation model. We compared the performance of an exact inference using an HMM using a discrete approximation of the models, with an approximate inference using an AMPF on the exact model. The simplified model leads to faster inference and a similar performance as the full model, with the performance extending to full length pieces and generalizing to different music styles. However, the proposed way to enrich the observation model did not lead to significant differences in performance. This might be caused by the simplistic audio features, and improving signal representations appears as a necessary next step. In the future, we plan to explore approximate inference in improved models (such as [13] using an improved state space discretization and tempo transition model) that also use better observation models and can effectively utilize multiple rhythmic patterns. We also plan to extend meter tracking to Hindustani music, where long cycles (longer than a minute) exist and hence present additional challenges.

Acknowledgments

This work is supported by the European Research Council (grant number 267583) and a Marie Curie Intra-European Fellowship (grant number 328379). The authors also thank Florian Krebs and Sebastian Böck at Johannes Kepler University, Linz, Austria for providing access to their code repositories.

6. REFERENCES

[1] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 602–607, Taipei, Taiwan, 2014.

[2] M. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Technical Report C4DM-09-06*, 2009.

[3] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 2009.

[4] S. Durand, J. P. Bello, B. David, and G. Richard. Downbeat tracking with multiple features and deep neural networks. In *Proc. of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

[5] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1832–1844, 2006.

[6] D. K. Grunberg, A. M. Batula, and Y. E. Kim. Towards the development of robot musical audition. In *Proc. of the Music, Mind, and Invention Workshop (MMI)*, New Jersey, USA, 2012.

[7] J. A. Hockman, M. E. P. Davies, and I. Fujinaga. One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 169–174, Porto, Portugal, October 2012.

[8] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, November 2012.

[9] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the "odd": Meter inference in a culturally diverse music corpus. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 425–430, Taipei, Taiwan, 2014.

[10] A. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics and Probability Letters*, 78(12):1498–1504, 2008.

[11] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the Meter of Acoustic Musical Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.

[12] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat- and downbeat tracking in musical audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 227–232, 2013.

[13] F. Krebs, S. Böck, and G. Widmer. An efficient state-space model for joint tempo and meter tracking. In *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, October 2015.

[14] F. Krebs, A. Holzapfel, A.T. Cemgil, and G. Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, May 2015.

[15] J. London. *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, Oxford, 2004.

[16] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–636, Utrecht, Netherlands, August 2010.

[17] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 19(6):1754–1769, 2011.

[18] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, February 1989.

[19] P. Sambamoorthy. *South Indian Music Vol. I-VI*. The Indian Music Publishing House, 1998.

[20] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In Search of Automatic Rhythm Analysis Methods for Turkish and Indian Art Music. *Journal of New Music Research*, 43(1):97–117, 2014.

[21] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proc. of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5237–5241, Florence, Italy, May 2014.

[22] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multimodality through mixture tracking. In *Proc. of the 9th IEEE International Conference on Computer Vision*, pages 1110–1116, Nice, France, October 2003.

[23] N. Whiteley, A. T. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, 2006.

[24] N. Whiteley, A. T. Cemgil, and S. Godsill. Sequential inference of rhythmic structure in musical audio. In *Proc. of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 1321–1325, Honolulu, USA, April 2007.