ON-LINE LEARNING FOR THE INFINITE HIDDEN MARKOV MODEL

Abel Rodriguez
Department of Applied Mathematics and Statistics
University of California
Santa Cruz, California, 95062
abel@ams.ucsc.edu

Key Words: Particle Learning, Nonparametric Bayes, Infinite Hidden Markov Model, Hierarchical Dirichlet Process.

ABSTRACT

We develop a sequential Monte Carlo algorithm for the infinite hidden Markov model (iHMM) that allows us to perform on-line inferences on both system states *and* structural (static) parameters. The algorithm described here provides a natural alternative to Markov chain Monte Carlo samplers previously developed for the iHMM, and is particularly helpful in applications where data is collected sequentially and model parameters need to be continuously updated. We illustrate our approach in the context of both a simulation study and a financial application.

1. INTRODUCTION

Hidden Markov models (HMMs) (Cappé et al., 2005) are an extremely popular class of models in statistics and machine learning, with applications in fields as diverse as language modeling, genomics and finance. HMMs are characterized by a sequence of unobserved states $\xi_1, \ldots, \xi_T$ where $\xi_t \in \{1, \ldots, L\}$, which are related through a Markov chain with transition probabilities $\mathsf{Pr}(\xi_t = j | \xi_{t-1} = i) = \pi_{ij}$, i.e., $\xi_t | \xi_{t-1} \sim \mathsf{Multinom}(\boldsymbol{\pi}_{\xi_{t-1}})$ with $\xi_0 \sim \mathsf{Multinom}(\boldsymbol{\pi}_0)$. Conditional on the states, the observations $\mathbf{y}_1, \ldots, \mathbf{y}_T$ are assumed to be independent and identically distributed, $\mathbf{y}_t | \xi_t \sim \psi(\mathbf{y}_t | \boldsymbol{\vartheta}_{\xi_t})$, where $\psi(\cdot | \boldsymbol{\vartheta}_{\xi_t})$ is the emission distribution, and $\boldsymbol{\vartheta}_l$ is the value of the emission parameter when the system is in state l.

One of the main challenges in the application of HMMs is selecting the right number of states;

models with too few states lack flexibility, while too many states might lead to overfitting and poor predictive performance. Infinite hidden Markov models (iHMMS) (Beal et al., 2001; Teh et al., 2006) solve this problem by allowing for an infinite number of states and treating the number of *active* states (those which have generated observations) as a random parameter to be estimated from the data. The iHMM we discuss in this paper was introduced by Teh et al. (2006), and is constructed as a hierarchical extension of the Dirichlet process mixture (DPM) model (Ferguson, 1973; Lo, 1984; Sethuraman, 1994; Escobar & West, 1995), which is one of the most popular approaches to nonparametric modeling in Bayesian statistics.

Bayesian learning algorithms for the iHMM based on Markov chain Monte Carlo samplers have been developed in Teh et al. (2006) and van Gael et al. (2008). However, MCMC algorithms, which iteratively sample blocks of model parameters conditional on the rest, are not well adapted to problems in which data is collected sequentially and parameters need to be continuously updated. In this context, sequential Monte Carlo (SMC) algorithms (Cappe et al., 2007) represent a faster and more efficient alternative. In their simpler form, SMC algorithms reduce to a sequential application of importance sampling. In importance sampling, samples simulated under an instrumental distribution are reweighed to obtain an approximation to the posterior distribution of interest. In particular, the algorithm we discuss in this paper is an extension of the particle learning (PL) approach discussed in Carvalho et al. (2008) and Carvalho et al. (2009). Although this is by no means the only class of SMC algorithms that allows for simultaneous learning on both the system states and the structural parameters controlling the evolution of the system, we focus on PL algorithms because they greatly simplifies the estimation of unknown static parameters.

The remaining of the paper is organized as follows: Sections 2 and 3 provide brief self-contained reviews of the Dirichlet process and the infinite Hidden Markov model, including a description of the two MCMC algorithms available for the iHMM: the collapsed Gibbs sampler Teh et al. (2006) and the beam sampler van Gael et al. (2008). Section 4 reviews the general particle learning approach introduced in Carvalho et al. (2008) and Carvalho et al. (2009), and Section 5 describes in detail its application to the iHMM. Section 6 presents two illustrations of the algorithm, a simulation study where the performance of the PL algorithm is compared against the performance of existing MCMC algorithms, and a stochastic volatility model for financial data. Finally, Section 7 presents a short discussion and future research directions.

2. DIRICHLET PROCESSES: BASICS AND NOTATION

A random distribution G is said to follow a Dirichlet process (DP) prior (Ferguson, 1973; Sethuraman, 1994) with baseline measure H and precision α, denoted $G \sim \mathsf{DP}(\alpha, H)$, if it can be represented as:

$$G(\cdot) = \sum_{l=1}^{\infty} \omega_l \delta_{\boldsymbol{\vartheta}_l}(\cdot)$$

where $\delta_a(\cdot)$ denotes the degenerate measure at a, $\boldsymbol{\vartheta}_l \sim H$ independently for every l, and $\omega_l = u_l \prod_{k<l}(1-u_k)$ with $u_l \sim \mathsf{Beta}(1, \alpha)$. Note that $\sum_{l=1}^{\infty} \omega_l = 1$ almost surely; in the sequel, we say that the vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots)$ follows a stick breaking distribution with parameter α, denoted $\mathsf{SB}(\alpha)$.

The Dirichlet process is most commonly used in the context of hierarchical models as a prior on the distribution of the model parameters, which leads to the Dirichlet process mixture model (Lo, 1984; Escobar, 1994; Escobar & West, 1995). In this case, the model for the observables $\mathbf{y}_1, \ldots, \mathbf{y}_T$ becomes

$$\mathbf{y}_t | \boldsymbol{\theta}_t \sim \psi(\mathbf{y}_t | \boldsymbol{\theta}_t) \qquad \boldsymbol{\theta}_t | G \sim G \qquad G \sim \mathsf{DP}(\alpha, H) \tag{1}$$

where, as before, $\psi(\cdot|\boldsymbol{\theta})$ is a parametric kernel indexed by $\boldsymbol{\theta}$. The Dirichlet process mixture model can also be obtained as the limit of a finite mixture model (Neal, 2000); consider the finite mixture

$$\mathbf{y}_t | \xi_t \sim \psi(\mathbf{y}_t | \boldsymbol{\vartheta}_{\xi_t}) \qquad \xi_t \sim \mathsf{Multinom}(\alpha/L, \ldots, \alpha/L) \tag{2}$$

with $\boldsymbol{\vartheta}_l \sim H$. As $L \to \infty$, the marginal distribution of $\mathbf{y}_t$ induced by (2) converges to the one induced by (1).

A sample $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ where $\boldsymbol{\theta}_t | G \sim G$ and $G \sim \mathsf{DP}(\alpha, H)$ can be characterized using a Pólya urn scheme (Blackwell & MacQueen, 1973). After integrating out the unknown distribution G, the joint distribution for $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_T$ can be described by a sequence of predictive distributions where $\boldsymbol{\theta}_1 \sim H$ and

$$\boldsymbol{\theta}_{t+1} | \boldsymbol{\theta}_t, \dots, \boldsymbol{\theta}_1 \sim \sum_{l=1}^{t} \frac{1}{\alpha + t} \delta_{\boldsymbol{\theta}_l} + \frac{\alpha}{\alpha + t} H \qquad\qquad 1 \leq t \leq T$$

This Pólya urn can be rewritten in terms of a collection of independent and identically distributed random variables $\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \dots$ and a sequence of indicators $\xi_1, \dots, \xi_T$ such that $\xi_1 = 1$ and

$$\xi_{t+1} | \xi_t, \dots, \xi_1 \sim \sum_{l=1}^{L_t} \frac{n_{lt}}{\alpha + t} \delta_l + \frac{\alpha}{\alpha + t} \delta_{L_t + 1} \qquad\qquad 1 \leq t \leq T,$$

where $n_{lt} = \sum_{i=1}^{t} \mathbf{1}(\xi_i = l)$ is the number of observations assigned to group l among the first t of them, and L_t is the number of unique values among the first t observations. The original sample $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_T$ is then obtained by setting $\boldsymbol{\theta}_t = \boldsymbol{\vartheta}_{\xi_t}$.

Due to exchangeability, this Pólya urn also describes the prior full conditional distribution for any $\boldsymbol{\theta}_t$, which is the basis to implement Gibbs sampling schemes for the Dirichlet process mixture model. From the Pólya urn it is clear that, although the DP has infinite capacity (i.e., an infinite number of components), for any finite sample of size T, at most T of them will be active (i.e., have observations assigned to them).

3. INFINITE HIDDEN MARKOV MODELS

Infinite hidden Markov models (iHMMs) (Beal et al., 2001; Teh et al., 2006) generalize HMMs to include an infinite number of states, in the same way the Dirichlet process model generalizes finite mixtures. This increases flexibility by providing a straightforward procedure to treat the number of active states L as a random variable, which is to be estimated from the data rather than fixed in advanced. Following Teh et al. (2006), we consider a model of the form

$$\mathbf{y}_t | \xi_t \sim \psi(\mathbf{y}_t | \boldsymbol{\vartheta}_{\xi_t}) \qquad \xi_t | \xi_{t-1} \sim \mathsf{Multinom}(\boldsymbol{\pi}_{\xi_{t-1}}) \qquad \boldsymbol{\pi}_l | \boldsymbol{\beta} \sim \mathsf{DP}(\alpha, \boldsymbol{\beta}) \qquad \boldsymbol{\beta} \sim \mathsf{SB}(\gamma)$$

where $\boldsymbol{\vartheta}_l \sim H$. Therefore, $\boldsymbol{\pi}_l$ is the vector transition probabilities out of state l and $\boldsymbol{\beta}$ corresponds to the average of these transition probabilities. As with DP mixutres, the iHMM includes in principle an infinite number of states; in practice, at most T (and usually much smaller number) are active, i.e., generate observations. The concentration parameters α and γ control $\mathsf{E}(L|T)$ the number of expected active states in the chain.

In the sequel, we assume that the emission distribution $\psi(\mathbf{y}|\boldsymbol{\vartheta})$ belongs to the exponential family and is indexed by the natural parameter $\boldsymbol{\vartheta} \in \boldsymbol{\Theta}$, i.e, ψ can be written as

$$\psi(\mathbf{y}|\boldsymbol{\vartheta}) = \exp\left\{\boldsymbol{\vartheta}'\mathbf{s}(\mathbf{y}) + c(\boldsymbol{\vartheta}) + q(\mathbf{y})\right\} \qquad \boldsymbol{\vartheta} \in \boldsymbol{\Theta} \qquad \mathbf{y} \in \mathcal{Y}$$

where the spaces $\boldsymbol{\Theta}$ and $\mathcal{Y}$ are independent of $\mathbf{y}$ and $\boldsymbol{\vartheta}$ respectively, $\mathbf{s}(\mathbf{y})$ is a (possibly multivariate-valued) function of $\mathbf{y}$ alone (the sufficient statistic associated with the likelihood ψ), and $c(\boldsymbol{\vartheta})$ and $q(\mathbf{y})$ are real valued functions of, respectively, $\boldsymbol{\vartheta}$ and $\mathbf{y}$ alone. We also assume that H is the conjugate distribution for $\boldsymbol{\vartheta}$, i.e., H has density h (or probability mass function h, if $\boldsymbol{\Theta}$ is a discrete space) given by:

$$h(\boldsymbol{\vartheta}|\boldsymbol{\eta}, \nu) = \exp\left\{\boldsymbol{\vartheta}'\boldsymbol{\eta} + \nu c(\boldsymbol{\vartheta}) + b(\nu, \boldsymbol{\eta})\right\}$$

where $\boldsymbol{\eta}$ and ν are the hyperparameters of the baseline measure, $c(\boldsymbol{\vartheta})$ is defined as above and $b(\nu, \boldsymbol{\eta})$ is a real valued function of ν and $\boldsymbol{\eta}$ alone. This setup encompasses a large number of broadly used emission distributions including the Gaussian, multinomial, Poisson and Gamma. To increase flexibility, we also incorporate priors for α and γ, so that $\alpha \sim \mathsf{Gam}(a_\alpha, b_\alpha)$ and $\gamma \sim \mathsf{Gam}(a_\gamma, b_\gamma)$.

A marginal Gibbs sampler for the iHMM that avoids explicitly representing the transition probabilities $\{\pi_l\}$ and the component-specific parameters $\{\boldsymbol{\vartheta}_l\}$ is discussed in Teh et al. (2006). Note that, after marginalization, the state parameters $\xi_1, \ldots, \xi_n$ can updated by sampling from their full conditional distribution, $\xi_t| \cdots \sim \mathsf{Multinom}(q_1^{-t}, \ldots, q_{L+1}^{-t})$, where

$$q_l^{-t} \propto \begin{cases} \frac{(n^{-t}_{\xi_{t-1},l} + \alpha\beta_l)(n^{-t}_{l,\xi_{t+1}} + \alpha\beta_{\xi_{t+1}})}{n_{l\cdot}^{-t} + \alpha} d_l^{-t} & l \leq L, l \neq \xi_{t-1} \\ \frac{(n^{-t}_{\xi_{t-1},l} + \alpha\beta_l)(n^{-t}_{l,\xi_{t+1}} + \alpha\beta_{\xi_{t+1}} + 1)}{n_{l\cdot}^{-t} + \alpha + 1} d_l^{-t} & l = \xi_{t-1} = \xi_{t+1} \\ \frac{(n^{-t}_{\xi_{t-1},l} + \alpha\beta_l)(n^{-t}_{l,\xi_{t+1}} + \alpha\beta_{\xi_{t+1}})}{n_{l\cdot}^{-t} + \alpha + 1} d_l^{-t} & l = \xi_{t-1} \neq \xi_{t+1} \\ \alpha\beta_l\beta_{\xi_{t+1}} d_{L+1}^{-t} & l = L + 1 \end{cases}$$

n_{ij}^{-t} denotes the number of transitions form state i to state j (excluding that those that involve time t), $n_{i\cdot}^{-t} = \sum_j n_{ij}^{-t}$ is the number of transitions out of state i, $n_{\cdot j}^{-t} = \sum_i n_{ij}^{-t}$ is the number of transitions into state j, and

$$d_l^{-t} = p(\mathbf{y}_t | \{\mathbf{y}_j : j \neq t, \xi_j = l\}) = \frac{\exp\left\{b(\nu + n_{l\cdot}, \mathbf{z}_l^{-t} + \boldsymbol{\eta})\right\}}{\exp\left\{b(\nu + n_{l\cdot} + 1, \mathbf{s}(\mathbf{y}_t) + \mathbf{z}_l^{-t} + \boldsymbol{\eta})\right\}}$$

for $l \leq L$ and

$$d_{L+1}^{-t} = p(\mathbf{y}_t) = \frac{\exp\left\{q(\mathbf{y}_t) + b(\nu, \boldsymbol{\eta})\right\}}{\exp\left\{b(\nu + 1, \mathbf{s}(\mathbf{y}_t) + \boldsymbol{\eta})\right\}}$$

In the above expressions $\mathbf{z}_l^{-t} = \sum_{\{j:j\neq t, \xi_j = l\}} \mathbf{s}(\mathbf{y}_j)$ is the (conditional) sufficient statistic for $\boldsymbol{\vartheta}_l$ excluding $\mathbf{y}_t$.

Given the states, the prior mean for the transition probabilities, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_L, \beta_{L+1})$, can be sampled by introducing independent auxiliary variables $\{m_{ij}\}$ for $i, j \in \{1, \ldots, L\}$ such that

$$\mathsf{Pr}(m_{ij} = m | \cdots) \propto S(n_{ij}, m)(\alpha\beta_j)^m \qquad\qquad m = 0, \ldots, n_{ij}$$

where $S(\cdot, \cdot)$ denotes the Stirling number of the first kind. Now, conditional on these auxiliary variables we can update $\boldsymbol{\beta}$ by sampling

$$\boldsymbol{\beta} | \cdots \sim \mathsf{Dir}(m_{\cdot 1}, \ldots, m_{\cdot L}, \gamma)$$

where $m_{\cdot j} = \sum_{i=1}^{L} m_{ij}$. Auxiliary variables also facilitate sampling for the shape parameters α and γ. To sample α we introduce auxiliary variables $\varsigma_i \sim \mathsf{Beta}(\alpha + 1, n_{i\cdot})$ and $u_i \sim \mathsf{Ber}(n_{i\cdot}/(\alpha + n_{i\cdot}))$ for $i = 1, \ldots, L$. Conditional on these latent variables, and assuming $\alpha \sim \mathsf{Gam}(a_\alpha, b_\alpha)$ a priori, we can update α from

$$\alpha | \cdots \sim \mathsf{Gam}\left(a_\alpha + m_{\cdot\cdot} - u_{\cdot}, b_\alpha - \sum_{i=1}^{L} \log \varsigma_i\right).$$

Finally, for γ, we introduce another auxiliary variable $\phi \sim \mathsf{Beta}(\gamma + 1, m_{\cdot\cdot})$. Under the gamma prior, the full conditional distribution for γ given ϕ corresponds to a mixture of two gamma distributions,

$$\gamma | \cdots \sim \epsilon\, \mathsf{Gam}(a_\gamma + L, b_\gamma - \log(\phi)) + (1 - \epsilon)\mathsf{Gam}(a_\gamma + L - 1, b_\gamma - \log(\phi)).$$

where $\epsilon/(1-\epsilon) = (a_\gamma + L - 1)/\{m_{\cdot\cdot}(b_\gamma - \log(\phi))\}$. Since this algorithm updates each ξ_t separately from its full conditional distribution, it leads to highly correlated posterior samples. As an alternative, van Gael et al. (2008) propose a slice sampler for the iHMM based on the algorithm of Walker (2007). By explicilty representing the transition probabilities $\{\boldsymbol{\pi}_l\}$ and introducing auxiliary variables that adaptively truncate the number states in the model, they are able to use a Forward-Backward algorithm to jointly sample the state variables, therefore reducing correlation in the Markov chain.

The two algorithms we just described are ideally suited for off-line problems where data is to be analyzed in batch. However, they have serious drawbacks in on-line applications where data needs to be processed sequentially. In that type of setting, a new run of the MCMC algorithm is necessary for each new observation, which might be prohibitively time consuming. Even in off-line problems, MCMC algorithms can be extremely inefficient if autocorrelations in the chain are very high. Since an iHMM defines a state space model, sequential Monte Carlo algorithms are a natural alternative to MCMC that can overcome their drawbacks.

4. SEQUENTIAL MONTE CARLO AND PARTICLE LEARNING: BASICS AND NOTATION

A state-space model is defined through a hierarchical specification

$$\mathbf{y}_t|\xi_t, \boldsymbol{\zeta} \sim \psi(\mathbf{y}_t|\xi_t, \boldsymbol{\zeta}) \qquad \xi_t|\xi_{t-1}, \boldsymbol{\zeta} \sim p(\xi_t|\xi_{t-1}, \boldsymbol{\zeta}) \qquad \xi_0|\boldsymbol{\zeta}, \sim p(\xi_0|\boldsymbol{\zeta}) \qquad \boldsymbol{\zeta} \sim p(\boldsymbol{\zeta}),$$

where $p(\xi_t|\xi_{t-1}, \boldsymbol{\zeta})$ is a transition distribution describing the evolution of the unobserved state, $p(\xi_0|\boldsymbol{\zeta})$ describes the distribution of the initial state and $p(\boldsymbol{\zeta})$ is the prior distribution on the vector of structural parameters $\boldsymbol{\zeta}$ that control the evolution and emission processes.

Two typical problems associated with learning in state space models are filtering and smoothing. In filtering problems we are interested in sequentially updating $p(\xi_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t)$ to obtain $p(\xi_{t+1}, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_{t+1})$, while in smoothing problems the goal is to obtain $p(\xi_1, \ldots, \xi_T, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_T)$. The classical example of recursive filtering and smoothing is the Kalman filter (Kalman, 1960), which gives a close form solution for linear Gaussian systems with known structural parameters. In more general settings, where closed form expressions are not available for the filtering and smoothing distributions, simulation algorithms that use discrete approximations based on a random cloud of particles have gained popularity in the last few years.

More concretely, sequential Monte Carlo (SMC) samplers use particle approximations of the form

$$\hat{p}^N(\xi_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t) = \sum_{i=1}^{N} w_t^{(i)} \delta_{(\hat{\xi}_t^{(i)}, \boldsymbol{\zeta}_t^{(i)})}(\xi_t, \boldsymbol{\zeta})$$

to represent $p(\xi_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t)$, and use importance sampling to sequentially update the distribution once a new observation is collected. The algorithms are constructed to ensure, convergence is distribution, i.e., $\hat{p}^N(\xi_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t) \xrightarrow{D} p(\xi_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t)$ as $N \to \infty$

The literature on sequential Monte Carlo methods is extensive, an excellent review can be found in Cappe et al. (2007). In this paper we will focus on the particle learning (PL) approach described in Carvalho et al. (2008) and Carvalho et al. (2009) because it simplifies the simultaneous estimation of states and structural parameters. The application of the PL framework assumes that, at any time t, the posterior distribution for $\boldsymbol{\zeta}$ depends on the states and observations through a low dimensional vector of sufficient statistics $\mathbf{r}_t$, which can be sequentially updated using a deterministic recursion $\mathcal{R}$ such that $\mathbf{r}_{t+1} = \mathcal{R}(\mathbf{r}_t, \mathbf{y}_{t+1}, \xi_{t+1})$, so that $p(\boldsymbol{\zeta}|\xi_1, \ldots, \xi_t, \mathbf{y}_1, \ldots, \mathbf{y}_t) = p(\boldsymbol{\zeta}|\mathbf{r}_t)$. PL also reqires that the predictive distribution $p(\mathbf{y}_{t+1}|\mathbf{x}_t, \boldsymbol{\zeta}) = \int p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, \boldsymbol{\zeta}) p(\mathbf{x}_{t+1}|\mathbf{x}_t, \boldsymbol{\zeta}) d\mathbf{x}_{t+1}$, with $\mathbf{x}_t = (\mathbf{r}_t, \xi_t)$, can be computed in closed form. If these two conditions are satisfied, we can treat the sufficient statistics $\mathbf{r}_t$ as deterministically updated states and write:

$$p(\mathbf{x}_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_{t+1}) \propto p(\mathbf{y}_{t+1}|\mathbf{x}_t, \boldsymbol{\zeta}) p(\mathbf{x}_t, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_t)$$

and from there

$$p(\mathbf{x}_{t+1}, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_{t+1}) = \int p(\boldsymbol{\zeta}|\mathbf{r}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t, \boldsymbol{\zeta}^*, \mathbf{y}_{t+1}) p(\mathbf{x}_t, \boldsymbol{\zeta}^*|\mathbf{y}_1, \ldots, \mathbf{y}_{t+1}) d\boldsymbol{\zeta}^* d\mathbf{x}_t$$

where

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \boldsymbol{\zeta}, \mathbf{y}_{t+1}) = p(\mathbf{r}_{t+1}|\mathbf{r}_t, \xi_{t+1}, \mathbf{y}_{t+1}) p(\xi_{t+1}|\xi_t, \boldsymbol{\zeta}, \mathbf{y}_{t+1})$$

and $p(\mathbf{r}_{t+1}|\mathbf{r}_t, \xi_{t+1}, \mathbf{y}_{t+1})$ a point mass concentrated in $\mathcal{R}(\mathbf{r}_t, \mathbf{y}_{t+1}, \xi_{t+1})$. These expressions lead to the following iterative algorithm to update the filtering distributions:

Algorithm 1 Particle learning filtering.

```
1. Sample $\hat{\boldsymbol{\zeta}}_0^{(i)} \sim p(\boldsymbol{\zeta})$ and $\hat{\xi}_0^{(i)} \sim p(\xi_0|\hat{\boldsymbol{\zeta}}_0^{(i)})$, and initialize $\hat{\mathbf{r}}_0^{(i)}$.
for $t = 0$ to $T - 1$ do
    2. Set $v_t^{(i)} = p(\mathbf{y}_{t+1}|\hat{\xi}_t^{(i)}, \hat{\mathbf{r}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)})$.
    3. Set $w_t^{(i)} = \frac{v_t^{(i)}}{\sum_{s=1}^N v_t^{(s)}}$ and sample $(\tilde{\xi}_t^{(i)}, \tilde{\mathbf{r}}_t^{(i)}, \tilde{\boldsymbol{\zeta}}_t^{(i)}) \sim \sum_{s=1}^N w_t^{(s)} \delta_{(\hat{\xi}_t^{(s)}, \hat{\mathbf{r}}_t^{(s)}, \hat{\boldsymbol{\zeta}}_t^{(s)})}$.
    4. Sample $\hat{\xi}_{t+1}^{(i)} \sim p(\xi_{t+1}|\tilde{\xi}_t^{(i)}, \tilde{\mathbf{r}}_t^{(i)}, \tilde{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_t)$.
    5. Update $\hat{\mathbf{r}}_{t+1}^{(i)} = \mathcal{R}(\tilde{\mathbf{r}}_t^{(i)}, \mathbf{y}_t, \hat{\xi}_{t+1}^{(i)})$.
    6. Sample $\hat{\boldsymbol{\zeta}}_{t+1}^{(i)} \sim p(\boldsymbol{\zeta}|\hat{\mathbf{r}}_{t+1}^{(i)})$.
end for
```

Note that, when referring to sampled particles, we slightly abuse notation by writing $\hat{\boldsymbol{\zeta}}_t$ to denote a particle for $\boldsymbol{\zeta}$ drawn conditional of the information available up to time t. Also, note that step 6 is not formally necessary, since the particles $\tilde{\boldsymbol{\zeta}}_t^{(1)}, \ldots, \tilde{\boldsymbol{\zeta}}_t^{(N)}$ were resampled conditionally on all the information available up to time $t + 1$. Instead, step 6 can be seen as an instance of MCMC adaptation that improves the performance of the SMC sampler by increasing particle diversity.

Once the filtering algorithm has been run for $t = 1, \ldots, T$, the stored particles representations for the marginal distributions $\{p(\mathbf{x}_t, \boldsymbol{\zeta}_t|\mathbf{y}_t, \ldots, \mathbf{y}_1)\}_{t=1}^T$ can be used to generated sample paths from the joint distribution $p(\mathbf{x}_T, \ldots, \mathbf{x}_1, \boldsymbol{\zeta}_t|\mathbf{y}_t, \ldots, \mathbf{y}_1)$ by using the smoothing algorithm described in Godsill et al. (2004). The algorithm is repeated for $b = 1, \ldots, B$ to generate B sample paths. In the specific case of PL, the aforementioned algorithm takes the following form:

Algorithm 2 Particle learning smoothing.

```
1. Sample $(\xi_T^{(b)}, \mathbf{r}_T^{(b)}, \boldsymbol{\zeta}^{(b)}) \sim p(\xi_T, \mathbf{r}_T, \boldsymbol{\zeta}|\mathbf{y}_1, \ldots, \mathbf{y}_T) = \sum_{i=1}^N \frac{1}{N} \delta_{(\hat{\xi}_T^{(i)}, \hat{\mathbf{r}}_T^{(i)}, \hat{\boldsymbol{\zeta}}_T^{(i)})}$
for $t = T - 1$ to $1$ do
    2. Set $q_t^{(i)} = p(\xi_{t+1}^{(b)}, \mathbf{r}_{t+1}^{(b)}|\hat{\xi}_t^{(i)}, \hat{\mathbf{r}}_t^{(i)}, \boldsymbol{\zeta}^{(b)})$.
    3. Set $\varpi_t^{(i)} = \frac{q_t^{(i)}}{\sum_{s=1}^N q_t^{(s)}}$ and sample $(\xi_t^{(b)}, \mathbf{r}_t^{(b)}) \sim \sum_{s=1}^N \varpi_t^{(i)} \delta_{(\hat{\xi}_t^{(s)}, \hat{\mathbf{r}}_t^{(s)})}$.
end for
```

5. PARTICLE LEARNING FOR THE INFINITE HIDDEN MARKOV MODEL

We proceed now to apply the PL framework in the context of the iHMM. As Teh et al. (2006), we explicitly integrate out the transition probabilities $\{\boldsymbol{\pi}_l\}$ and the state-specific parameters $\{\boldsymbol{\vartheta}_l\}$. Our approach relies on the fact that, once the transition probabilities $\{\boldsymbol{\pi}_l\}$ has been integrated out of the model, the transition distribution can be written as:

$$p(\xi_{t+1}|\xi_t, \dots, \xi_1, \boldsymbol{\beta}, \alpha) = \sum_{l=1}^{L_t} \frac{n_{\xi_t l t} + \alpha\beta_l}{n_{\xi_t \cdot t} + \alpha}\delta_k + \frac{\alpha\beta_{L_t+1}}{n_{\xi_t \cdot t} + \alpha}\delta_{L_t+1} \tag{3}$$

where $L_t = \max\{\xi_1, \dots, \xi_t\}$ is the number of distinct states visited by the process up to time t, n_{ijt} is the number of transitions between states i and j up to time t, and $n_{i \cdot t}$ is the number of transitions out of state i up to time t. As a consequence, the one step ahead predictive distribution reduces to

$$\begin{aligned} p(\mathbf{y}_{t+1}|\xi_t, \mathbf{r}_t, \boldsymbol{\beta}, \alpha) = {} & \frac{\alpha\beta_{L_t+1}}{n_{\xi_t \cdot t} + \alpha} \exp\left\{q(\mathbf{y}_t) + b(\nu, \boldsymbol{\eta}) - b(\nu + 1, \mathbf{s}(\mathbf{y}_{t+1}) + \boldsymbol{\eta})\right\} \\ & + \sum_{l=1}^{L_t} \frac{(n_{\xi_t l t} + \alpha\beta_l)\exp\left\{q(\mathbf{y}_t) + b(\nu + n_{\cdot lt}, \mathbf{z}_{lt} + \boldsymbol{\eta})\right\}}{(n_{\xi_t \cdot t} + \alpha)\exp\left\{b(\nu + n_{\cdot lt} + 1, \mathbf{s}(\mathbf{y}_{t+1}) + \mathbf{z}_{lt} + \boldsymbol{\eta})\right\}} \end{aligned} \tag{4}$$

where $\mathbf{z}_{lt} = \sum_{\{j: j \le t, \xi_j = l\}} \mathbf{s}(\mathbf{y}_j)$ is the sufficient statistic associated with the emission parameters of component l at time t, and $\mathbf{r}_t = (L_t, \{n_{ijt}\}, \{\mathbf{z}_{lt}\})$ is the vector of sufficient statistics for the problem. In order to be able to update the structural parameters $\boldsymbol{\beta}$, α and γ, we augment the particle system by explicitly sampling the structural parameters and the auxiliary variables we described in the context of MCMC algorithms for the iHMM in Section 3. Therefore, for the iHMM

$$(\hat{\mathbf{x}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)}) = (\hat{\xi}_t^{(i)}, \hat{L}_t^{(i)}, \{\hat{n}_{ljt}^{(i)}\}, \{\hat{\mathbf{z}}_{lt}^{(i)}\}, \hat{\alpha}_t^{(i)}, \{\hat{\beta}_{lt}^{(i)}\}, \hat{\gamma}_t^{(i)}, \{\hat{m}_{ijt}^{(i)}\}, \hat{\phi}_t^{(i)}, \{\hat{u}_{it}^{(i)}\}, \{\hat{\varsigma}_{it}^{(i)}\})$$

and Algorithm 1 becomes:

1. Compute weights $w_t^{(i)} = v_t^{(i)} / \sum_{j=1}^{N} v_t^{(j)}$, where $v_t^{(i)} = \sum_{l=1}^{\hat{L}_t^{(i)}+1} q_l^{(i)}(\hat{\mathbf{x}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1})$ and

$$q_l^{(i)}(\hat{\mathbf{x}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1}) = \frac{(\hat{n}^{(i)}_{\hat{\xi}_t^{(i)} lt} + \hat{\alpha}_t^{(i)}\hat{\beta}_l^{(i)})\exp\left\{b(\nu + \hat{n}^{(i)}_{\cdot lt}, \mathbf{z}^{(i)}_{lt} + \boldsymbol{\eta})\right\}}{(\hat{n}^{(i)}_{\hat{\xi}_t^{(i)} \cdot t} + \hat{\alpha}_t^{(i)})\exp\left\{b(\nu + \hat{n}^{(i)}_{\cdot lt} + 1, \mathbf{s}(\mathbf{y}_{t+1}) + \mathbf{z}^{(i)}_{lt} + \boldsymbol{\eta})\right\}}$$

for $l \leq \hat{L}_t^{(i)}$ and

$$q^{(i)}_{\hat{L}_t^{(i)}+1}(\hat{\mathbf{x}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1}) = \frac{\hat{\alpha}_t^{(i)} \hat{\beta}^{(i)}_{\hat{L}_t^{(i)}+1} \exp\{b(\nu, \boldsymbol{\eta})\}}{(\hat{n}^{(i)}_{\hat{\xi}_t^{(i)} \cdot t} + \tilde{\alpha}_t^{(i)}) \exp\{b(\nu+1, \mathbf{s}(\mathbf{y}_{t+1}) + \boldsymbol{\eta})\}}$$

2. Sample $(\tilde{\mathbf{x}}_t^{(1)}, \tilde{\boldsymbol{\zeta}}_t^{(1)}), \ldots, (\tilde{\mathbf{x}}_t^{(N)}, \tilde{\boldsymbol{\zeta}}_t^{(N)})$ from

$$\tilde{p}(\mathbf{x}_t, \boldsymbol{\zeta} | \mathbf{y}_1, \ldots, \mathbf{y}_{t+1}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{(\hat{\mathbf{x}}_t^{(i)}, \hat{\boldsymbol{\zeta}}_t^{(i)})}(\mathbf{x}_t, \boldsymbol{\zeta})$$

3. Propagate the particles to generate $(\hat{\mathbf{x}}_{t+1}^{(i)}, \boldsymbol{\zeta}_{t+1}^{(i)})$ by:

   (a) Sampling $\hat{\xi}_{t+1}^{(i)}$ from $p(\hat{\xi}_{t+1}^{(i)} | \tilde{\xi}_t^{(i)}, \tilde{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1})$, where

$$p(\hat{\xi}_{t+1}^{(i)} | \tilde{\xi}_t^{(i)}, \tilde{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1}) \propto \sum_{l=1}^{\hat{L}_t^{(i)}+1} \frac{q_l^{(i)}(\tilde{\mathbf{x}}_t^{(i)}, \tilde{\boldsymbol{\zeta}}_t^{(i)}, \mathbf{y}_{t+1})}{\sum_{j=1}^{\hat{L}_t^{(i)}+1} q_l^{(j)}(\tilde{\mathbf{x}}_t^{(j)}, \tilde{\boldsymbol{\zeta}}_t^{(j)}, \mathbf{y}_{t+1})} \delta_l(\hat{\xi}_{t+1}^{(i)})$$

   (b) Updating the number of states by setting

$$\hat{L}_{t+1}^{(i)} = \begin{cases} \tilde{L}_t^{(i)} + 1 & \hat{\xi}_{t+1}^{(i)} = \tilde{L}_t^{(i)} + 1 \\ \hat{L}_{t+1}^{(i)} = \tilde{L}_t^{(i)} & \text{otherwise} \end{cases}$$

   (c) Updating the sufficient statistics by setting

$$\hat{n}^{(i)}_{\tilde{\xi}_t, \hat{\xi}_{t+1}, t+1} = \tilde{n}^{(i)}_{\tilde{\xi}_t, \hat{\xi}_{t+1}, t} + 1 \qquad \hat{\mathbf{z}}^{(i)}_{\hat{\xi}_{t+1}, t+1} = \tilde{\mathbf{z}}^{(i)}_{\hat{\xi}_{t+1}, t} + \mathbf{s}(\mathbf{y}_t)$$

   (d) If $\hat{\xi}_{t+1}^{(i)} \leq \tilde{L}_t^{(i)}$ set $\hat{\boldsymbol{\beta}}_{t+1}^{(i)} = \tilde{\boldsymbol{\beta}}_t^{(i)}$. Otherwise, update the transition probability vector by setting

$$\hat{\beta}_{l,t+1}^{(i)} = \begin{cases} \tilde{\beta}_{lt}^{(i)} & l < \hat{L}_{t+1}^{(i)} \\ \varphi \tilde{\beta}^{(i)}_{\tilde{L}_t^{(i)}+1,t} & l = \hat{L}_{t+1}^{(i)} \\ (1-\varphi) \tilde{\beta}^{(i)}_{\tilde{L}_t^{(i)}+1,t} & l = \hat{L}_{t+1}^{(i)} + 1 \end{cases}$$

   where $\varphi \sim \mathsf{Beta}(1, \tilde{\gamma}_t^{(i)})$.

4. Resample the structural parameters and auxiliary variables by:

   (a) Sampling $\hat{m}^{(i)}_{l,j,t+1} \in \{0, \dots, \hat{n}^{(i)}_{l,j,t+1}\}$ with

   $$\mathsf{Pr}(\hat{m}^{(i)}_{l,j,t+1} = m) \propto S(\hat{n}^{(i)}_{l,j,t+1}, m)(\tilde{\alpha}^{(i)}_t \hat{\beta}^{(i)}_{j,t+1})^m$$

   (b) Sampling $\hat{\gamma}^{(i)}_{t+1}$ by first sampling $\hat{\phi}^{(i)}_{t+1} \sim \mathsf{Beta}(\tilde{\gamma}^{(i)}_t + 1, \hat{m}^{(i)}_{\cdot\cdot t+1})$ and then sampling $\hat{\gamma}^{(i)}_{t+1}$ from

   $$\begin{aligned}\hat{\gamma}^{(i)}_{t+1} \sim \epsilon^{(i)}_{t+1}\mathsf{Gam}(a_\gamma + \hat{L}^{(i)}_{t+1}, b_\gamma - \log(\hat{\phi}^{(i)}_{t+1}))+ \\ (1 - \epsilon^{(i)}_{t+1})\mathsf{Gam}(a_\gamma + \hat{L}^{(i)}_{t+1} - 1, b_\gamma - \log(\hat{\phi}^{(i)}_{t+1})\end{aligned}$$

   where

   $$\frac{\epsilon^{(i)}_{t+1}}{1 - \epsilon^{(i)}_{t+1}} = \frac{a_\gamma + \hat{L}^{(i)}_{t+1} - 1}{\hat{m}^{(i)}_{\cdot\cdot t+1}(b_\gamma - \log(\hat{\phi}^{(i)}_{t+1}))}.$$

   (c) Sampling $\hat{\alpha}^{(i)}_{t+1}$ by first generating (for $l = 1, \dots, \hat{L}^{(i)}_{t+1}$)

   $$\hat{\varsigma}^{(i)}_{l,t+1} \sim \mathsf{Beta}(\tilde{\alpha}^{(i)}_t + 1, \hat{n}^{(i)}_{l\cdot t+1})$$

   and

   $$\hat{u}^{(i)}_{l,t+1} \sim \mathsf{Ber}(\hat{n}^{(i)}_{l\cdot t+1}/(\tilde{\alpha}^{(i)}_t + \hat{n}^{(i)}_{l\cdot t+1}))$$

   and then

   $$\hat{\alpha}^{(i)}_{t+1} \sim \mathsf{Gam}\left(a_\alpha + \hat{m}^{(i)}_{\cdot\cdot t+1} - \hat{u}^{(i)}_{\cdot t+1}, b_\alpha - \sum_{l=1}^{\hat{L}^{(i)}_{t+1}} \log \hat{\varsigma}^{(i)}_{l,t+1}\right)$$

   (d) Resampling

   $$\hat{\boldsymbol{\beta}}^{(i)}_{t+1} \sim \mathsf{Dir}(\hat{m}_{\cdot,1,t+1}, \dots, \hat{m}_{\cdot,\hat{L}^{(i)}_{t+1},t+1}, \hat{\gamma}^{(i)}_{t+1})$$

The algorithm is initialized by sampling $\hat{\gamma}^{(i)}_0 \sim \mathsf{Gam}(a_\gamma, b_\gamma)$ and $\hat{\alpha}^{(i)}_0 \sim \mathsf{Gam}(a_\alpha, b_\alpha)$, setting $\hat{L}^{(i)}_0 = 0$ (which implies $w^{(i)}_0 \propto 1$, $\hat{L}^{(i)}_1 = 1$, $\hat{n}^{(i)}_{111} = 1$, $\mathbf{z}^{(i)}_{11} = \mathbf{s}(\mathbf{y}_1)$ and $\hat{m}^{(i)}_{111} = 1$ for all particles), and applying steps 4(b) to 4(d) above.

If only on-line inference is required, the filtering procedure described above produces the required sequence of filtered distributions. As observations are collected, the particles can be updated at a computational cost $o(N)$, which is much smaller than the cost of rerunning an MCMC for the same number of iterations (which would be $o(NT)$). If a sample from the smoothed distribution $p(\xi_1, \ldots, \xi_T | \mathbf{y}_1, \ldots, \mathbf{y}_T)$ is required, Algorithm 2 can be applied through the following steps:

1. Sample $(\xi_T^{(b)}, L_T^{(b)}, \{n_{ljT}^{(b)}\}, \alpha^{(b)}, \{\beta_l^{(b)}\}) \sim \sum_{s=1}^N \frac{1}{N} \delta_{(\hat{\xi}_T^{(i)}, \hat{L}_T^{(i)}, \{\hat{n}_{ljT}^{(i)}\}, \hat{\alpha}_T^{(i)}, \{\hat{\beta}_{lT}^{(i)}\})}$.

2. Sequentially sample $\xi_t^{(b)}$ by

   (a) Setting $q_t^{(i)} = \sum_{l=1}^{\hat{L}_t} \frac{\hat{n}_{\hat{\xi}_t^{(i)} lt} + \alpha^{(b)} \beta_l^{(b)}}{\hat{n}_{\hat{\xi}_t^{(i)} \cdot t} + \alpha^{(b)}} \delta_k(\xi_{t+1}^{(b)}) + \frac{\alpha^{(b)} \beta_{L_t+1}^{(b)}}{\hat{n}_{\hat{\xi}_t^{(i)} \cdot t} + \alpha^{(b)}} \delta_{\hat{L}_t+1}(\xi_{t+1}^{(b)})$.

   (b) Setting $\varpi_t^{(i)} = \frac{q_t^{(i)}}{\sum_{s=1}^N q_t^{(s)}}$ and sample

$$(\xi_t^{(b)}, L_t^{(b)}, \{n_{ljt}^{(b)}\}, \alpha^{(b)}, \{\beta_l^{(b)}\}) \sim \sum_{s=1}^N \varpi_t^{(i)} \delta_{(\hat{\xi}_t^{(i)}, \hat{L}_t^{(i)}, \{\hat{n}_{ljt}^{(i)}\}, \hat{\alpha}_t^{(i)}, \{\hat{\beta}_{lt}^{(i)}\})}.$$

This procedure is repeated B times to generate B independent sample paths. However, a drawback of the algorithm described above is that paths for $\xi_1, \ldots, \xi_T$ are restricted to the histories of the particles surviving up to time T, i.e., $q_t^{(i)} > 0$ only if $\hat{\xi}_{t+1}^{(i)}$ originated by propagating $\hat{\xi}_t^{(i)}$ in the filtering step.

To improve the diversity of the particles, we can decouple the paths from the histories by explicitly sampling the transition probabilities $\{\boldsymbol{\pi}_l\}$ and emission parameters $\{\boldsymbol{\vartheta}_l\}$ conditionally on the sufficient statistics of the problem at time T. Indeed, conditionally on L_T, the iHMM becomes a regular HMM; the transition probabilities satisfy

$$\boldsymbol{\pi}_l | \mathbf{y}_1, \ldots, \mathbf{y}_T \sim \mathsf{Dir}(\alpha + \hat{n}_{l \cdot T}, (\beta_1 + \hat{n}_{l1T}, \ldots, \beta_{L_T} + \hat{n}_{lL_T T}))$$

and the emission parameters can be sampled independently from its posterior distribution,

$$p(\boldsymbol{\vartheta}_l | \mathbf{y}_1, \ldots, \mathbf{y}_T) = \exp\left\{ \boldsymbol{\vartheta}'(\mathbf{z}_{lT} + \boldsymbol{\eta}) + (n_{\cdot lT} + \nu) c(\boldsymbol{\vartheta}) + b(n_{\cdot lT} + \nu, \mathbf{z}_{lT} + \boldsymbol{\eta}) \right\}$$

Given the transition and emission probabilities of a finite HMM, sampling of the states can now proceed using a standard Forward-Backward algorithm. The resulting algorithm follows through the following steps:

1. Sample $(\xi_T^{(b)}, L_T^{(b)}, \{n_{ljT}^{(b)}\}, \{\mathbf{z}_{lT}^{(b)}\}, \alpha^{(b)}, \{\beta_l^{(b)}\}) \sim \sum_{s=1}^{N} \frac{1}{N}\delta_{(\hat{\xi}_T^{(i)}, \hat{L}_T^{(i)}, \{\hat{n}_{ljT}^{(i)}\}, \{\hat{\mathbf{z}}_{lT}^{(i)}\}, \hat{\alpha}_T^{(i)}, \{\hat{\beta}_{lT}^{(i)}\})}$.

2. Sample $\{\boldsymbol{\pi}_l^{(b)}\}$ from

   $$\boldsymbol{\pi}_l^{(b)} \sim \mathsf{Dir}(\alpha^{(b)} + \hat{n}_{l\cdot T}^{(b)}, (\beta_1^{(b)} + \hat{n}_{l1T}^{(b)}, \ldots, \beta_{L_T^{(b)}}^{(b)} + \hat{n}_{lL_T^{(b)}T}^{(b)})) \qquad l = 1, \ldots, L_T^{(b)}$$

   (Note that $\boldsymbol{\pi}_l^{(b)}$ is of length $L_t^{(b)}$.)

3. Sample $\{\boldsymbol{\vartheta}_l^{(b)}\}$ from

   $$p(\boldsymbol{\vartheta}_l | \mathbf{y}_1, \ldots, \mathbf{y}_T) = \exp\left\{\boldsymbol{\vartheta}'(\mathbf{z}_{lT}^{(b)} + \boldsymbol{\eta}) + (n_{\cdot lT}^{(b)} + \nu)c(\boldsymbol{\vartheta}) + b(n_{\cdot lT}^{(b)} + \nu, \mathbf{z}_{lT}^{(b)} + \boldsymbol{\eta})\right\}$$

4. Sample $\xi_1^{(b)}, \ldots, \xi_T^{(b)}$ using a single Forward-Backward iteration that uses $\{\boldsymbol{\pi}_l^{(b)}\}$ and $\{\boldsymbol{\vartheta}_l^{(b)}\}$ as the parameters for the transitions and emission distributions.

It is worth emphasizing that, although by sampling $\{\boldsymbol{\pi}_l\}$, $\{\boldsymbol{\vartheta}_l\}$ we increase the execution time of the algorithm and the Monte Carlo error of the estimates, we also increases particle diversity and minimize the risk of particle degeneracy in the smoothing step.

The algorithms described above can be easily generalized to accommodate uncertainty in the hyperparameters $\boldsymbol{\eta}$ and ν, as well as non-conjugate prior distributions. Given a prior $p(\boldsymbol{\eta}, \nu)$, the hyperparameters of the baseline measure can be sampled by augmenting the state vector with samples from $\{\boldsymbol{\vartheta}_l\}$, which in turn can be used to sample $\boldsymbol{\eta}$ and ν form their full conditional distribution. Similarly, augmenting the state vector with $\{\boldsymbol{\vartheta}_l\}$ allows us to avoid an intractable integral when computing the filtering weights $\{q_l^{(i)}\}$ for a non-conjugate baseline measure H.

6. ILLUSTRATIONS

First, we concentrate on a simulation study similar to the one discussed in Teh et al. (2006) and van Gael et al. (2008). Data was generated form a negatively autocorrelated HMM with 4 states and a multinomial emission distribution with 8 possible categories. The transition and emission matrices T and E are given by:

$$T = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix} \qquad E = \begin{pmatrix} 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \end{pmatrix}$$

Twenty sequences of 500 characters were generated, and iHMMs fitted to each one of them using the collapsed Gibbs sampler of Teh et al. (2006) (GIBBS), the slice sampler of van Gael et al. (2008) (SLICE), and our particle learning algorithm (PL). Hyperpriors were set to $\alpha \sim \mathsf{Gam}(4, 2)$ $\gamma \sim \mathsf{Gam}(3, 6)$ for all models. Performance of the samplers was investigated by computing one-step-ahead predictive densities,

$$p(\mathbf{y}_{t+1}|\mathbf{y}_1, \ldots, \mathbf{y}_t) = \int p(\mathbf{y}_{t+1}|\xi_t, \boldsymbol{\beta}, \alpha)p(\xi_t, \boldsymbol{\beta}, \alpha|\mathbf{y}_1, \ldots, \mathbf{y}_t)d\xi_t d\boldsymbol{\beta} d\alpha$$

for $t = 451, \ldots, 500$. One-step ahead predictive densities provide a measure of the quality of the models explored by the samplers that incorporate information about the prior distributions. The log predictives for PL were computed based on $N = 5,000$ particles, while for both GIBSS and SLICE they were computed using 5,000 iterations of the algorithm obtained after a burn-in period of 2,000 iterations. Our simulations suggest that the predictive performance of PL is comparable to that of SLICE, and that both of them explore better models than GIBBS. Overall computation time for PL was roughly two orders of magnitude smaller than the time for GIBBS or SLICE because both algorithm had to be re-run at each time t. When inference is performed only for the whole sequence, execution time for PL is about 30% longer than for for GIBBS, and about 15% longer than SLICE.

Our second illustration involves a data set of weekly returns for the S&P500 index covering the ten-year period between April 21, 1997 and April 9, 2007, for a total of 520 observations. We have previously analyzed the same data set in Rodriguez et al. (2010). The top panel in Figure 1 shows the evolution of these returns. The series does not exhibit any long term trend, but different levels of volatility can be clearly seen. In particular, two slightly different regimes are apparent; before May 2003, periods of high-volatility are relatively frequent, while after May 2003, we can appreciate longer low-volatility periods.

We model this data using an iHHM where the observations (weekly log returns) follow a normal distribution with zero mean and unknown variance (we denote this model iHMM-SV). The baseline measure H is an inverse-Gamma distribution with mean 0.000492 (which implies that the annualized mean volatility is around 16%, a historically reasonable value), and 2 degrees of freedom. The precision parameters for the iHMM α and γ are both given independent Gamma priors, $\alpha \sim \mathsf{Gam}(1,1)$ and $\gamma \sim \mathsf{Gam}(1,1)$. The particle learning algorithm described in this paper was used to fit the model, with the posterior mean of the filtered volatilities plotted in the bottom panel of Figure 1. In order to provide a comparison, we also fitted a first-order autoregressive stochastic volatility model, as described in Jacquier et al. (1994) (we denote this model as AR1-SV) using the auxiliary particle filter with structural learning developed in Liu & West (2001). Hyperparameter for the AR1-SV were chosen consistently with the choice for the iHMM-SV. The corresponding posterior means of the filtered volatilities are also shown in the bottom panel of Figure 1.

The iHMM detects between 6 and 9 volatility states, with the mode being six states (posterior probability 0.47). In terms of the volatility estimates, the results from both models are qualitatively similar; both detect the presence of volatility spikes in mid 2000, late 2001 and late 2002, as well as the low volatility regime arising after 2004. However, the results are quantitatively different. Most of the time, the estimates from the iHMM-SV are more stable (in the sense that, most of the time, they show lower short-term volatility of volatility). However, when a spike is detected, the volatility estimates for the iHMM-SV are much larger than those from the AR1-SV. This is consistent with our experience with stochastic volatility models: when compared with continuous mixtures, countable mixtures tend to generate smoother estimates for high probability states and less smooth estimates for low probability estates.

7. DISCUSSION

The particle learning algorithm described in this paper is particularly appealing for applications where efficient on-line Bayesian learning in iHHM models is necessary. For this type of applications, it provides a much faster and efficient alternative to existing MCMC algorithms. However, even in off-line applications, the PL approach is competitive with MCMC algorithms and provides some important advantages. For example, PL algorithms provide a straightforward approximation to the marginal likelihood of the iHMM model, allowing us to perform model selection. This angle will be explored in more detail elsewhere.

Although we focused our attention on the simplest version of the iHMMs where the baseline measure H is conjugate to the emission distribution ψ, this is not a key feature of the algorithm. Extensions to models with non-conjugate baseline measures where already discussed at the end of Section 5. Similarly, state-persistent iHMMs as the ones discussed in Fox et al. (2008) can be easily analyzed.

`Matlab` code implementing the algorithms used for the illustrations in this paper can be obtained from the author. Besides releasing this code as `Matlab` and `R` libraries, our short term research plans include exploring the application of particle learning algorithms to other hierarchical nonparametric models such as the nested Dirichlet process Rodriguez et al. (2008), where a Pólya urn is available but is not helpful in constructing Gibbs sampling algorithms.

Acknowledgements

This work was partially supported by grant NSF-DMS 0915272. I would like to thank Matthew Taddy for helpful comments on the manuscript.

References

BEAL, M. J., GHAHRAMANI, Z. & RASMUSSEN, C. E. (2001). The infinite hidden markov model. In *Proceedings of Fourteenth Annual Conference on Neural Information Processing Systems*.

BLACKWELL, D. & MACQUEEN, J. B. (1973). Ferguson distribution via Pólya urn schemes. *The Annals of Statistics* **1**, 353–355.

CAPPE, O., GODSILL, S. J. & MOULINES, E. (2007). An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE* **95**, 899–924.

CAPPÉ, O., MOULINES, E. & RYDEN, T. (2005). *Inference in Hidden Markov Models*. Springer.

CARVALHO, C. M., JOHANNES, M., LOPES, H. F. & POLSON, N. G. (2008). Particle learning and smoothing. Technical report, Department of Statistical Sciences - Duke University.

CARVALHO, C. M., LOPES, H. F., POLSON, N. G. & TADDY, M. (2009). Particle learning for general mixtures. Technical report, Department of Statistical Sciences - Duke University.

ESCOBAR, M. D. (1994). Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association* **89**, 268–277.

ESCOBAR, M. D. & WEST, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of American Statistical Association* **90**, 577–588.

FERGUSON, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics* **1**, 209–230.

FOX, E., SUDDERTH, E., JORDAN, M. I. & WILLSKY, A. (2008). An hdp-hmm for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

VAN GAEL, J., SAATCI, Y., TEH, Y.-W. & GHAHRAMANI, Z. (2008). Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

GODSILL, S. J., DOUCET, A. & WEST, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association* **99**, 156–168.

JACQUIER, E., POLSON, N. G. & ROSSI, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of business and Economic Statistics* **12**, 371–389.

KALMAN, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering* **82**, 35–45.

LIU, J. & WEST, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice*, Eds. A. Doucet, N. de Freitas & N. Gordon. Springer.

LO, A. Y. (1984). On a class of Bayesian nonparametric estimates: I. Density estimates. *Annals of Statistics* **12**, 351–357.

NEAL, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics* **9**, 249–.

RODRIGUEZ, A., DUNSON, D. B. & GELFAND, A. E. (2008). The nested Dirichlet process, with Discussion. *Journal of American Statistical Association* **103**, 1131–1154.

RODRIGUEZ, A., DUNSON, D. B. & GELFAND, A. E. (2010). Latent stick-breaking processes. *Journal of the American Statistical Association* **105**, 647–659.

SETHURAMAN, J. (1994). A constructive definition of dirichelt priors. *Statistica Sinica* **4**, 639–650.

TEH, Y. W., JORDAN, M. I., BEAL, M. J. & BLEI, D. M. (2006). Sharing clusters among related groups: Hierarchical Dirichlet processes. *Journal of the American Statistical Association* **101**, 1566–1581.

WALKER, S. G. (2007). Sampling the dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation* **36**, 45–54.

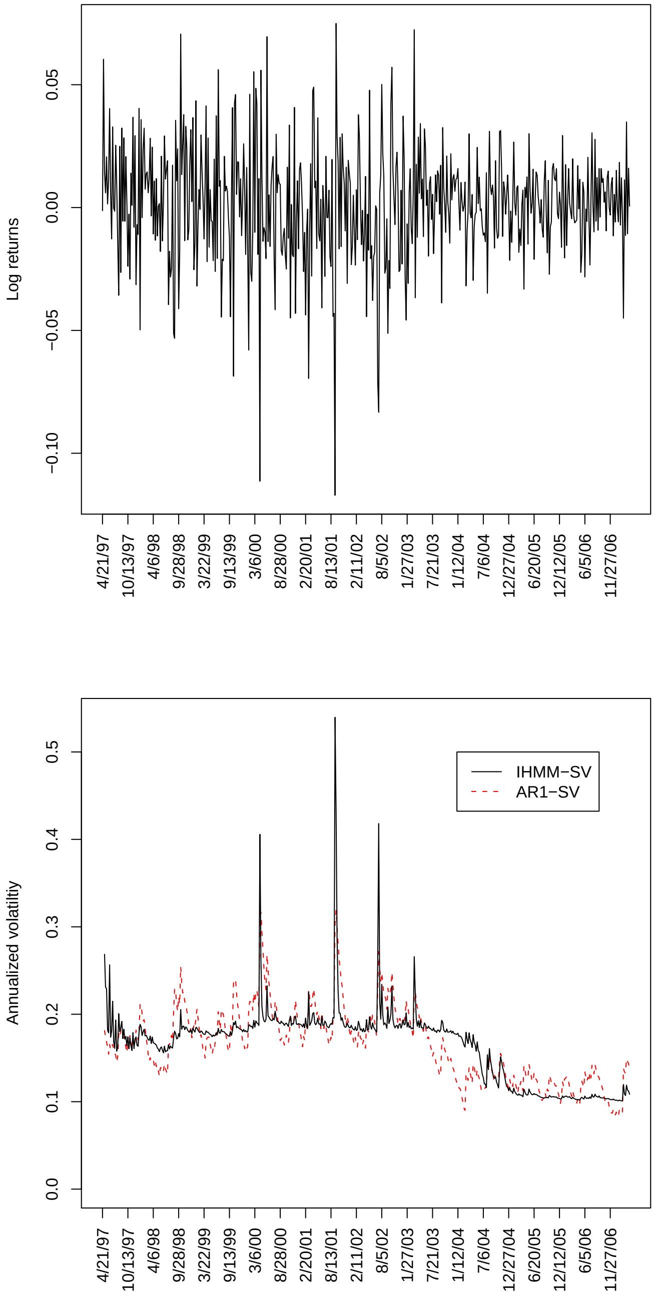

Figure 1: Raw data (left panel) and filtered volatilities (right panel) for the S&P500 data set.