On particle-based online smoothing and parameter inference in general hidden Markov models

Johan Westerborn

TRITA-MAT-A 2015:06
ISRN KTH/MAT/A-15/06-SE
ISBN 978-91-7595-554-4

Department of Mathematics
Royal Institute of Technology
SE-100 44, Stockholm
Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen, fredagen den 5 juni 2015 klockan 14.00 i rum 3721, Institutionen för matematik, Lindstedtsvägen 25, Kungl Tekniska Högskolan, Stockholm.

Printed in Stockholm by Universitetsservice US-AB

Abstract

This thesis consists of two papers studying online inference in general hidden Markov models using sequential Monte Carlo methods.

The first paper present an novel algorithm, the particle-based, rapid incremental smoother (PaRIS), aimed at efficiently perform online approximation of smoothed expectations of additive state functionals in general hidden Markov models. The algorithm has, under weak assumptions, linear computational complexity and very limited memory requirements. The algorithm is also furnished with a number of convergence results, including a central limit theorem.

The second paper focuses on the problem of online estimation of parameters in a general hidden Markov model. The algorithm is based on a forward implementation of the classical expectation-maximization algorithm. The algorithm uses the PaRIS algorithm to achieve an efficient algorithm.

Sammanfattning

Denna avhandling består av två artiklar som behandlar inferens i dolda Markovkedjor med generellt tillståndsrum via sekventiella Monte Carlo-metoder.

Den första artikeln presenterar en ny algoritm, PaRIS, med målet att effektivt beräkna partikelbaserade online-skattningar av utjämnade väntevärden av additiva tillståndsfunktionaler. Algoritmen har, under svaga villkor, en beräkningkomplexitet som växer endast linjärt med antalet partiklar samt högst begränsade minneskrav. Dessutom härleds ett antal konvergensresultat för denna algoritm, såsom en central gränsvärdessats.

Den andra artikeln fokuserar på online-estimering av modellparametrar i en generella dolda Markovkedjor. Den presenterade algoritmen kan ses som en kombination av PaRIS och en nyligen föreslagen online-implementation av den klassiska EM-algoritmen.

List of papers

This thesis consists of two papers referred to in the text with the capital letters A and B.

Paper A. *Efficient particle-based online smoothing in general hidden Markov models: the PaRIS algorithm*, Jimmy Olsson and Johan Westerborn

Paper B. *An efficient particle-based online EM algorithm for general state-space models*, Jimmy Olsson and Johan Westerborn

Acknowledgments

I would like to thank my supervisor Associate Professor Jimmy Olsson for all his constant support and encouragement.

I would also like to thank all of my colleagues at the Division of Mathematical Statistics, especially my fellow PhD students, for all their support.

Lastly I wish to thank Therese for her endless love and patience.

1. Overview

In this thesis we consider the problem of inference and parameter estimation in general *hidden Markov models* (HMMs), alternatively termed general *state space-models* (SSMs), using *sequential Monte Carlo* (SMC) methods, also known as *particle filters*.

An HMM consists of two processes: a hidden *Markov chain*, referred to as the *state process*, which is only partially observed through an *observation process* in such a way that conditionally on the states, the observations are statistically independent with conditional distribution of each observation depending on the corresponding state only. The generality of HMMs allows these to be applied in many different contexts and research fields, from economics [8] to speech recognition [29] and computational biology [27] (the over 300 references in [4] gives an idea of the applicability of these models). When working with HMMs we are typically given a fixed record, or data-stream, of observations from the observation process and are interested in calculating conditional distributions of one or several states given these observations. As we will see, the estimation of such conditional distributions will be particularly critical for parameter calibration.

In an HMM, the state process may, e.g., be the location and velocity of a moving target and the observation process the bearing of this target with respect to an observation point. We are then interested in estimating the most likely position and velocity of the target given all the observations up to a given time (see [20]). In this case the distribution of interest is the *filter distribution*, i.e., the conditional law of the current state given the observations up to the current time point. However, if we wish to know the most likely trajectory of this target, focus us set on the *smoothing distribution*, i.e., the conditional joint law of all latent states up to the current time point given the corresponding observations.

There are only two classes of HMMs that allow for exact computation of the filter and smoothing distributions, namely when the state space is finite (in which case the solution is provided by the *Baum-Welch algorithm* [30]) or when the model is linear and Gaussian (using *disturbance smoothing* [3, 26], which is closely related to the *Kalman filter* [24]). For nonlinear models it is possible obtain approximate solutions using the *extended Kalman filter* [1], where the model is linearized around an estimate of the mean and covariance using a first order Taylor expansion. Nevertheless, if the model is highly nonlinear, the extended Kalman filter has generally poor performance. To some extent, this can be counteracted by the *unscented Kalman filter* [23], which recover the filter mean and variance by propagating, using a deterministic sampling technique, a set of so-called “sigma points” through the nonlinear functions.

In the light of the previous, we are, in the presence of nonlinear and/or non-Gaussian model components, generally referred to finding approximations of the filter and smoothing distributions. In this thesis we will use SMC methods for this purpose. These methods, also known as particle methods, propagate, through a series of sampling and resampling steps, particles and weights such that the weighted empirical measure associated with the sample provides a discrete approximation of the distributions of interest. We refer to [16, 7, 17, 2, 6] for introductions to the topic. In a practical application, we require the algorithm under consideration to be numerically efficient in the sense that the computational time of the algorithm should grow only slowly with the number of particles; moreover, the algorithm should be numerically stable in the sense that the Monte Carlo variance should be controlled in the long-term. This is feasible in the case of filtering (see [15, 12, 9] for theoretical results on the time uniform convergence of particle filters), while estimation of the smoothing distribution flow poses generally more of a problem. Standard methods for estimation of smoothed expectations requires typically the data to be processed in two passes, in the forward as well as the backward directions [19, 17, 18], resulting in *batch algorithms* that have to be re-executed if a new observation becomes available. Another problem with current algorithms (see, e.g., [11]) is that the computational complexity is growing typically quadratically with the number of particles, resulting in slow algorithms.

In Paper A, a novel algorithm, *the particle-based, rapid incremental smoother* (PaRIS), for efficient online approximation of smoothed expectation of *additive state functionals* in general HMMs is presented. As we will see in the next section, the computation of smoothed expectations of additive form is crucial for parameter inference in HMMs. The algorithm, which processes the data in a single forward sweep, has a computational complexity that grows only linearly with the number of particles used. The PaRIS algorithm is coupled with several convergence results including a Hoeffding type inequality and a central limit theorem.

In Paper B, the ideas of the PaRIS algorithm is used for designing an efficient online *expectation-maximization* algorithm for parameter estimation in HMMs. Like PaRIS, this algorithm has a computational complexity that grows only linearly with the number of particles.

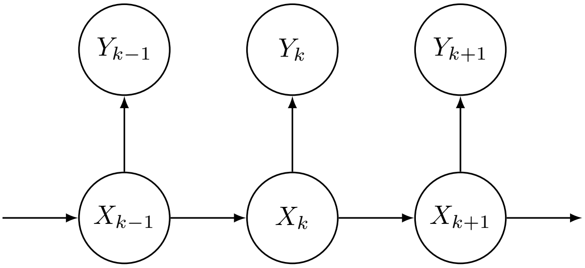

FIGURE 1. Hidden Markov model

2. Hidden Markov models

As mentioned previously, an HMM comprises two processes: a state process $\{X_t\}_{t\in\mathbb{N}}$ and an observation process $\{Y_t\}_{t\in\mathbb{N}}$. These two processes can be described by the following dynamical system:

$$\begin{aligned} X_{t+1} \mid X_t = x_t &\sim q_\theta(x_t, \cdot), \\ Y_t \mid X_t = x_t &\sim g_\theta(x_t, \cdot), \end{aligned}$$

where q_θ and g_θ are Markov transition densities, the latter referred to as the *emission density*, and θ is a vector containing model parameters. The Markov chain $\{X_t\}_{t\in\mathbb{N}}$, which is unobserved, is initialized according to some initial distribution χ. The dependence structure of an HMM can be described by the graphical model in Figure 1. For a rigorous treatment of HMMs, we refer to [7, Section 2].

Example 1 (Stochastic volatility)**.** *As an example of an HMM, consider the following nonlinear stochastic volatility model [21]:*

$$\begin{aligned} X_{t+1} &= \phi X_t + \sigma\varepsilon_{t+1} \\ Y_t &= \beta\exp(X_t/2)\zeta_t \end{aligned} \quad (t \in \mathbb{N}), \tag{2.1}$$

where $\{\varepsilon_t\}_{t\in\mathbb{N}}$ and $\{\zeta_t\}_{t\in\mathbb{N}}$ are sequences of mutually independent standard normally distributed random variables. The strength of this model is that it allows volatility clustering of the asset log-returns Y, and the parameter ϕ can be interpreted as the persistence in the volatility shocks. Figure 2 displays a trajectory generated through simulation under the parameter vector $\theta = (\phi, \beta^2, \sigma^2) = (.9, .3^2, .9^2)$.

In the following we assume that we are given a fixed sequence $\{y_t\}_{t\in\mathbb{N}}$ of observations of $\{Y_t\}_{t\in\mathbb{N}}$. Any kind of statistical inference in HMMs involves

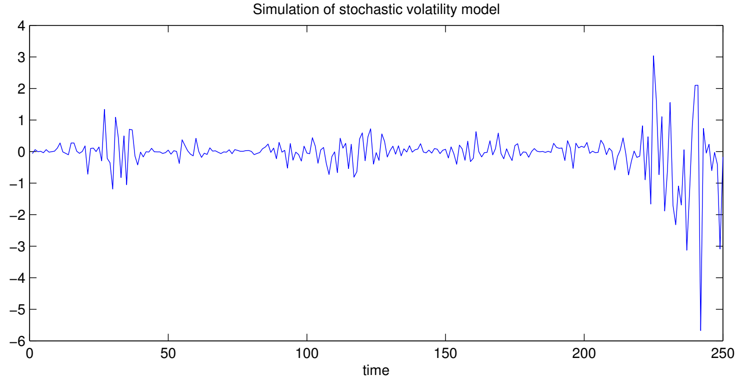

FIGURE 2. Simulated values of $Y_{0:250}$ using the parameters $\phi = .9$, $\beta^2 = .3^2$, and $\sigma^2 = .9^2$.

typically the computation of conditional distributions of the state process, or parts of it, given the observations. We let $\phi_{s:s'|t;\theta}h := \mathbb{E}_\theta[h(X_{s:s'}) \mid Y_{0:t}]$ denote the expectation of a function of the hidden states given all observations. Using Bayes's formula, this expectation can be written as

$$\phi_{s:s'|t;\theta}h = \frac{\int \cdots \int h(x_{s:s'}) g_\theta(x_0, y_0)\, \chi(x_0) \prod_{\ell=0}^{t-1} g_\theta(x_{\ell+1}, y_{\ell+1}) q_\theta(x_\ell, x_{\ell+1}) \,\mathrm{d}x_{0:t}}{\int \cdots \int g_\theta(x_0, y_0) \chi(x_0) \prod_{\ell=0}^{t-1} g_\theta(x_{\ell+1}, y_{\ell+1}) q_\theta(x_\ell, x_{\ell+1}) \,\mathrm{d}x_{0:t}},$$

where we assumed that the denominator is positive. Of particular interest are the distributions $\phi_{t;\theta} := \phi_{t:t|t;\theta}$ and $\phi_{0:t;\theta} := \phi_{0:t|t;\theta}$, i.e., the filter and smoothing distributions, respectively.

An interesting feature of an HMM is that conditionally on the observations, the state process is still a Markov chain in the forward as well as the time-reversed directions. The conditional density of X_t given $X_{t+1} = x_{t+1}$ and the data $y_{0:t}$, which we refer to as the *backward transition density*, is denoted by $\overleftarrow{q}_{\phi_{t;\theta}}(x_{t+1}, x_t)$. Using the backward density it is possible to rewrite the smoothing distribution as

$$\phi_{0:t;\theta}h = \int \cdots \int \left(\prod_{s=0}^{t-1} \overleftarrow{q}_{\phi_{s;\theta}}(x_{s+1}, x_s) \right) \phi_{t;\theta}(x_t) h(x_{0:t}) \,\mathrm{d}x_{0:t}. \qquad (2.2)$$

The previous expression is known as the *backward decomposition* of the smoothing distribution.

As will soon become apparent, one is most often interested in estimating $\phi_{0:t;\theta} h_t$ for h_t being of additive form

$$h_t(x_{0:t}) = \sum_{\ell=0}^{t-1} \tilde{h}_\ell(x_{\ell:\ell+1}). \tag{2.3}$$

Before using an HMM in practice, e.g., for prediction, the unknown model parameters θ have to be calibrated, and such parameter estimation involves generally the computation of smoothed expectations over objective functions of type (2.3). One way of calibrating the parameters goes via the *maximum likelihood method*, which returns the parameter estimate $\hat{\theta} = \operatorname{argmax}_\theta \ell(\theta; y_{0:t})$ maximizing the *log-likelihood function* $\ell(\theta; y_{0:t})$ (i.e., the logarithm of the joint density of the observations $y_{0:t}$ as a function of θ). For an HMM, the log-likelihood is given by

$$\ell(\theta; y_{0:t}) := \log \left(\int \cdots \int g_\theta(x_0, y_0) \chi(x_0) \prod_{\ell=0}^{t-1} g_\theta(x_{\ell+1}, y_{\ell+1}) q_\theta(x_\ell, x_{\ell+1}) \, \mathrm{d}x_{0:t} \right),$$

which, again, is intractable for most models. As it is generally the case for latent data models, it is easier is to consider instead the *complete data log-likelihood* $\ell(\theta; y_{0:t}, x_{0:t})$ defined as the logarithm of the joint density of the observations as well as the hidden states as a function of the parameter, i.e.,

$$\begin{aligned} \ell(\theta; y_{0:t}, x_{0:t}) &:= \log \left(g_\theta(x_0, y_0) \chi(x_0) \prod_{\ell=0}^{t-1} g_\theta(x_{\ell+1}, y_{\ell+1}) q_\theta(x_\ell, x_{\ell+1}) \right) \\ &= \log \chi(x_0) + \sum_{\ell=1}^{t} \log q_\theta(x_{\ell-1}, x_\ell) + \sum_{\ell=0}^{t} \log g_\theta(x_\ell, y_\ell). \end{aligned}$$

The complete data log-likelihood depends also on the latent data $X_{0:t}$, which has to be estimated. The *expectation-maximization* (EM) *algorithm* (presented in [13]) consists roughly in repeating this procedure until the parameters converge. More specifically, the EM algorithm introduces the *intermediate quantity*

$$\mathcal{Q}(\theta, \theta^i) := \mathbb{E}_{\theta^i} \left[\ell(\theta; Y_{0:t}, X_{0:t}) \mid Y_{0:t} \right],$$

where $\mathbb{E}_{\theta^i}$ denotes expectation under the model parametrized by some given parameter estimate θ^i. The parameter estimate is then updated recursively as $\theta^{i+1} = \operatorname{argmax}_\theta \mathcal{Q}(\theta, \theta^i)$. Calculating $\mathcal{Q}(\theta, \theta^i)$ is referred to as the *expectation-step* (or *E-step*) and finding the maximum is referred to as the *maximization-step* (or *M-step*). The EM algorithm is initialized by setting θ^0 arbitrarily. The convergence of this algorithm was, under certain assumptions, established in [31].

The EM algorithm is particularly easy to implement if the distribution of the complete data $(X_{0:t}, Y_{0:t})$ belongs to an *exponential family*, i.e., the complete data log-likelihood can be written as

$$\ell(\theta; y_{0:t}, x_{0:t}) = \langle \phi(\theta), s_t(x_{0:t}) \rangle - c(\theta) + \log h(x_{0:t}),$$

where $\langle \cdot, \cdot \rangle$ denotes scalar product, $\phi(\theta)$ and $c(\theta)$ are known functions, and $s_t(x_{0:t})$ are some (possibly vector-valued) sufficient statistics. In this case, The E-step then boils down to computing $\phi_{0:t;\theta} s_t$, and maximizing $\mathcal{Q}(\theta, \theta^i)$ is reduced to maximizing $\langle \phi(\theta), \phi_{0:t;\theta} s_t \rangle - c(\theta)$ with respect to θ. This maximization should allow a closed form solution $\theta^{i+1} = \Lambda(\phi_{0:t;\theta} s_t)$, which is often the case. Observe that we should calculate the smoothed expectation of the sufficient statistics under the model dynamics determined by the current parameter estimate θ^i. As mentioned earlier, the smoothing distribution lacks closed-form solution in the general case and we are hence referred to finding approximations of the smoothed expectations under consideration. This standard version of EM, where the whole trajectory $y_{0:t}$ is processed from scratch before each parameter update, is referred to as the *batch* EM algorithm.

Example 2 (Stochastic volatility, cont.). *The model in Example 1 comprises four sufficient statistics of additive form* (2.3)*, with terms given by*

$$\tilde{s}_t^1(x_{t:t+1}) = x_t^2, \qquad \tilde{s}_t^2(x_{t:t+1}) = x_t x_{t+1},$$
$$\tilde{s}_t^3(x_{t:t+1}) = x_{t+1}^2, \qquad \tilde{s}_t^4(x_{t:t+1}) = y_{t+1}^2 \exp(-x_{t+1}).$$

For this model, the M-step involves the update

$$\Lambda(z_1, z_2, z_3, z_4) = \left(\frac{z_2}{z_3}, z_3 - \frac{z_2^2}{z_1}, z_4 \right).$$

In Figure 3 the batch EM algorithm is used for estimating the parameters of the stochastic volatility model, on the basis of the trajectory in Figure 2. By taking the mean of the last 10 parameter estimates we obtain $\theta^* = (.917, .242^2, .904^2)$.

3. Particle methods for filtering and smoothing

SMC methods is a class of genetic-type algorithms that update sequentially a random sample of particles with associated importance weights. The weighted empirical measures formed by the particle sample at the different time steps serve as approximations of the filter distribution flow.

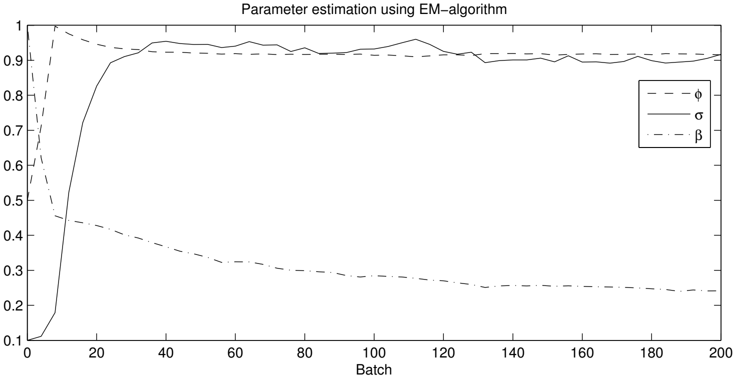

FIGURE 3. Parameter estimation in the stochastic volatility model using the EM algorithm. The trajectory simulated in Figure 2 is used as input. Taking the mean over the last 10 iterates yields $\theta^* = (.917, .242^2, .904^2)$.

3.1. **Particle filter.** The bootstrap particle filter, presented first in [20], constructs a sequence of weighted particles targeting the filter distribution flow $\{\phi_t\}_{t\in\mathbb{N}}$ given the sequence $\{y_t\}_{t\in\mathbb{N}}$ of observations. In order to describe the most basic SMC algorithm we proceed recursively and assume that we have at hand a weighted sample $\{(\omega_t^i, \xi_t^i)\}_{i=1}^N$ approximating the filter distribution $\phi_{t;\theta}$ in the sense that

$$\phi_{t;\theta}^N h = \sum_{i=1}^N \frac{\omega_t^i}{\Omega_t} h(\xi_t^i),$$

where $\Omega_t = \sum_{\ell=1}^N \omega_t^\ell$, converges to $\phi_{t;\theta}h$ as N tends to infinity. To form a weighted particle sample $\{(\omega_{t+1}^i, \xi_{t+1}^i)\}_{i=1}^N$ targeting the next filter $\phi_{t+1;\theta}$, we begin by resampling the current particles by drawing a vector $\{I_{t+1}^i\}_{i=1}^N$ of indices from the multinomial distribution with probabilities proportional to the weights, i.e.,

$$I_{t+1}^i \sim \mathsf{Pr}\left(\{\omega_t^\ell\}_{\ell=1}^N\right),$$

where $\mathsf{Pr}(\{\omega_t^\ell\}_{\ell=1}^N)$ refers to the discrete probability distribution induced by the weights $\{\omega_t^\ell\}_{\ell=1}^N$. After this, the particles are propagated through the model dynamics by, first, drawing $\xi_{t+1}^i \sim q_\theta(\xi_t^{I_{t+1}^i}, \cdot)$ and, second, assigning

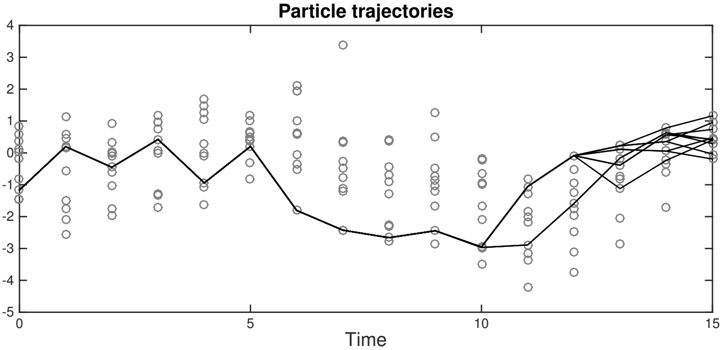

FIGURE 4. The historical trajectories of the particles when running the standard particle filter. The grey circles represent the particles and the black line the particles active in the estimate at time 15.

each particle the weight $\omega_{t+1}^i = g_\theta(\xi_{t+1}^i, y_{t+1})$. The algorithm is initialized by drawing the initial particles ξ_0^i from the initial distribution χ and letting the associated importance weight be $\omega_0^i = g_\theta(\xi_0^i, y_0)$.

As a by-product, the historical trajectories of the particle filter provides jointly an estimate of the joint smoothing distribution. These trajectories are constructed by linking each generation of particles with the corresponding ancestors. Since the ancestors are selected through resampling, this method suffers from a well-known degeneracy phenomenon that is seen in the collapse of the trajectories. In Figure 4, displaying a simulation of this phenomenon, we see clearly that for most time steps only a single particle contributes to the resulting estimator, which, consequently, will degenerate. See [28, 6, 6] for further discussion.

3.2. **Forward-filtering backward-smoothing.** To combat the particle path degeneracy, the backward decomposition (2.2) can be used, granted that we can approximate the backward transition densities $\overleftarrow{q}_{\phi_{s;\theta}}$. As approximations of these transition densities (on the supporting points formed by the particles) we consider

$$\overleftarrow{q}_{\phi^N_{s;\theta}}(\xi^i_{s+1}, \xi^j_s) = \frac{\omega^j_s q_\theta(\xi^j_s, \xi^i_{s+1})}{\sum_{\ell=1}^N \omega^\ell_s q_\theta(\xi^\ell_s, \xi^i_{s+1})}. \tag{3.1}$$

The *forward-filtering backward-smoothing* (FFBSm) *algorithm* [17, 22, 25] consists in simply inserting this approximation into (2.2), yielding the approximation

$$\phi^N_{0:t;\theta} h := \sum_{i_0=1}^N \cdots \sum_{i_t=1}^N \left(\prod_{s=0}^{t-1} \frac{\omega_s^{i_s} q_\theta(\xi_s^{i_s}, \xi_{s+1}^{i_{s+1}})}{\sum_{\ell=1}^N \omega_s^\ell q_\theta(\xi_s^\ell, \xi_{s+1}^{i_{s+1}})} \right) \frac{\omega_t^{i_t}}{\Omega_t} h(\xi_0^{i_0}, \ldots, \xi_t^{i_t}),$$

of $\phi_{0:t;\theta} h$. For a general objective function h, this occupation measure is impractical as the cardinality of its support grows geometrically fast with time. In the case of an objective function of additive form the computational complexity is still quadratic in the number of particles since the normalized constant $\sum_{\ell=1}^N \omega_s^\ell q_\theta(\xi_s^\ell, \xi_{s+1}^{i_{s+1}})$ needs to be computed for each particle and time point.

3.3. **Forward-only implementation of FFBSm.** As noted in [10], estimates of the sequence $\{\phi_{0:t;\theta} h_t\}_{t \in \mathbb{N}}$ may, in the case of objective functions of additive form (2.3), be computed recursively. This is done by noting that the sequence of functions $\mathbf{T}_t h_t(x_t) = \mathbb{E}[h_t(X_{0:t}) \mid X_t = x_t, y_{0:t}]$, $t \in \mathbb{N}$, may be updated recursively according to

$$\begin{aligned} &\mathbf{T}_{t+1} h_{t+1}(x_{t+1}) && (3.2) \\ &= \mathbb{E}[h_{t+1}(X_{0:t+1}) \mid X_{t+1} = x_{t+1}, y_{0:t+1}] \\ &= \mathbb{E}[\tilde{h}_t(X_{t:t+1}) + \mathbb{E}[h_t(X_{0:t}) \mid X_t = x_t, y_{0:t}] \mid X_{t+1} = x_{t+1}, y_{0:t+1}] \\ &= \int \overleftarrow{q}_{\phi_{t;\theta}}(x_{t+1}, x_t) \left(\tilde{h}_t(x_{t:t+1}) + \mathbf{T}_t h_t(x_t) \right) \mathrm{d}x_t. && (3.3) \end{aligned}$$

Each expectation of interest is then calculated as $\phi_{0:t;\theta} h_t = \phi_{t;\theta} \mathbf{T}_t h_t$. Plugging the approximations (3.1) into the recursion we get approximations $\{\tilde{\tau}_t^i\}_{i=1}^N$ of the statistics $\{\mathbf{T}_t h_t(\xi_t^i)\}_{i=1}^N$, initialized by $\tilde{\tau}_0^i = 0$. These approximations may be updated sequentially by first evolving the particle filter one step and then setting

$$\tilde{\tau}_{t+1}^i = \sum_{j=1}^N \frac{\omega_t^j q_\theta(\xi_t^j, \xi_{t+1}^i)}{\sum_{\ell=1}^N \omega_t^\ell q_\theta(\xi_t^\ell, \xi_{t+1}^i)} \left(\tilde{\tau}_t^j + \tilde{h}_t(\xi_t^j, \xi_{t+1}^i) \right), \tag{3.4}$$

yielding the estimate

$$\phi^N_{0:t+1;\theta} h_{t+1} = \sum_{i=1}^N \frac{\omega_{t+1}^i}{\Omega_{t+1}} \tilde{\tau}_{t+1}^i$$

of $\phi_{0:t+1;\theta} h_{t+1}$. This allows for online estimation. In addition, the algorithm has the appealing property that only the current statistics and particle sample need to be stored in the memory. The computational complexity of this algorithm is however still quadratic since a sum of N terms has to be computed for each particle [11].

3.4. **Forward-filtering backward-simulation.** In order to remedy the high computational complexity of FFBSm, the *forward-filtering backward-simulation* (FFBSi) generates trajectories on the *index space* of the particle locations generated by the particle filter. This is done by simulating repeatedly a time-reversed, inhomogeneous Markov chain with transition probabilities equal to the particle approximation of the backward densities, i.e., drawing indices $\{\tilde{J}_s\}_{s=0}^t$ with

$$\tilde{J}_t \sim \mathsf{Pr}(\{\omega_t^\ell\}_{\ell=1}^N),$$

and, recursively,

$$\tilde{J}_s \mid \tilde{J}_{s+1} = j \sim \mathsf{Pr}(\{\omega_s^i q_\theta(\xi_s^i, \xi_{s+1}^j)\}_{i=1}^N).$$

Given $\{\tilde{J}_s\}_{s=0}^t$, an approximate draw from the smoothing distribution is formed by the random vector $(\xi_0^{\tilde{J}_0}, \ldots, \xi_t^{\tilde{J}_t})$. Consequently, the uniformly weighted occupation measure associated with a set of conditionally independent such draws provides a finite-dimensional approximation of the smoothing distribution $\phi_{0:t;\theta}$; see [19]. In this basic formulation of FFBSi, the backward sampling pass requires the normalizing constants of the particle-based backward densities to be computed, and hence the algorithm suffers from a quadratic complexity. On the other hand, on the contrary to FFBSm, this complexity is the same for *all* types of objective functions.

However, following [14] it is, under the assumption that the transition density q_θ is bounded from above by some constant $\bar{\varepsilon}$, possible to reduce the computational complexity of FFBSi by simulating using the following accept-reject technique. In order to simulate $\tilde{J}_s$ given $\tilde{J}_{s+1} = j$ a candidate J^* is drawn from the proposal distribution $\mathsf{Pr}(\{\omega_s^i\}_{i=1}^N)$ is accepted with probability $q_\theta(\xi_s^{J^*}, \xi_{s+1}^j)/\bar{\varepsilon}$. The procedure is repeated until acceptance. Under the additional assumption that the transition density is bounded also from below it can be shown (see [14, Proposition 2]) that the expected number of proposals is bounded by a time independent constant, resulting in an algorithm that has only a linear computational complexity.

4. Summary of Paper A

Requiring separate forward and backward processing of the data, the standard design of FFBSi is impracticable in online applications. In Paper A, we propose a novel algorithm, the PaRIS algorithm, which can be viewed as a hybrid between the forward-only implementation of FFBSm and the FFBSi algorithm.

Similarly to the forward-only implementation of FFBSm, the PaRIS algorithm propagates particle estimates $\{\tau_t^i\}_{i=1}^N$ of $\{\mathbf{T}_t h_t(\xi_t^i)\}_{i=1}^N$. The key ingredient of PaRIS is the replacement of (3.4), in the spirit of FFBSi, by a Monte Carlo estimate. More specifically, for each particle location ξ_{t+1}^i we draw $\tilde{N}$ indices $\{J_{t+1}^{(i,j)}\}_{j=1}^{\tilde{N}}$ according to

$$J_{t+1}^{(i,j)} \sim \mathsf{Pr}\left(\{\omega_t^\ell q_\theta(\xi_t^\ell, \xi_{t+1}^i)\}_{\ell=1}^N\right). \tag{4.1}$$

and update the associated auxiliary statistic according to

$$\tau_{t+1}^i = \tilde{N}^{-1} \sum_{j=1}^{\tilde{N}} \left(\tau_t^{J_{t+1}^{(i,j)}} + \tilde{h}_t(\xi_t^{J_{t+1}^{(i,j)}}, \xi_{t+1}^i) \right).$$

At each time step, an estimator of $\phi_{0:t+1} h_{t+1}$ is obtained as $\sum_{i=1}^N \omega_{t+1}^i \tau_{t+1}^i / \Omega_{t+1}$. The algorithm is initialized by setting $\tau_0^i = 0$. In order to gain efficiency, (4.1) is performed using the accept-reject technique speeding up the FFBSi algorithm. This results in an algorithm having linear computational complexity.

In Paper A it is shown that the design parameter $\tilde{N}$ should be at least 2 in order for the algorithm to be numerically stable. Indeed, when $\tilde{N} = 1$, a phenomenon similar to the collapsing of the historical trajectories of the particle filter is observed. Figure 5 displays an illustration of the difference between the cases $\tilde{N} = 1$ and $\tilde{N} = 2$, where the similarity between the cases $\tilde{N} = 1$ and Figure 4 is visible. Paper A provides a full theoretical study of PaRIS, which is highly non-trivial due to the complex dependence structure induced by the backward sampling approach.

5. Summary of Paper B

In Paper B, an online EM algorithm for parameter estimation in hidden Markov models is presented. Applying the EM algorithm to HMMs belonging to the exponential family involves estimating $t^{-1}\phi_{0:t;\theta} s_t$ where s_t are (possibly vector-valued) sufficient statistics. Since these sufficient statistics are of additive form we can use (3.3) to achieve an online smoothing algorithm of the time-averaged smoothed sufficient statistics using the PaRIS algorithm. In the context of EM, the quantity of interest is generally the normalized expectation $t^{-1}\phi_{0:t;\theta} s_t$ rather than $\phi_{0:t;\theta} s_t$, and we hence propagate $t^{-1}\mathbf{T}_t s_t$

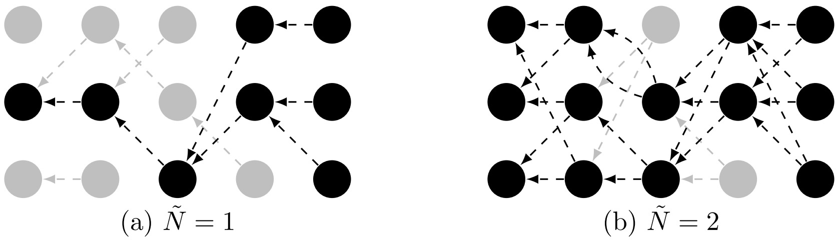
(a) $\tilde{N} = 1$ (b) $\tilde{N} = 2$

FIGURE 5. Genealogical traces corresponding to backward simulation in the PaRIS algorithm. Columns of nodes refer to different particle populations (with $N = 3$) at different time points (with time increasing rightward) and arrows indicate connections through the backward simulation in the PaRIS algorithm. Black-coloured particles are included in the final estimator, while grey-coloured ones are inactive.

according to the recursion

$$\gamma_t \mathbf{T}_t s_t(x_t) = \int \overleftarrow{q}_{\phi_{t-1;\theta}}(x_t, x_{t-1}) \big\{ (1-\gamma_t)\gamma_{t-1}\mathbf{T}_{t-1}s_{t-1}(x_{t-1}) \\ + \gamma_t \tilde{s}_{t-1}(x_{t-1:t}) \big\} \mathrm{d}x_{t-1}, \qquad (5.1)$$

where $\gamma_t = t^{-1}$. The main idea of the online EM algorithm is to update sequentially $t^{-1}\phi_{0:t}s_t$ through the recursion (5.1) and then use the maximization function Λ to update the parameters; see [5, 11]. In practice, instead of using $\gamma_t = t^{-1}$, a step size γ_t is used that satisfies the regular stochastic approximation requirements $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$. A usual choice is $\gamma_t = t^{-\alpha}$, where $.5 < \alpha \leq 1$.

Again exact computations of $\gamma_t \mathbf{T}_t s_t(x_t)$ is infeasible for most models, and we hence need to, as before, approximate this quantity. In Paper B this approximation is formed by applying the same technique as in the PaRIS algorithm, using accept-reject sampling to draw a sample from the backward kernel and taking the Monte Carlo average as an estimate. Thus, we propagate in parallel $\{\tau_t^i\}_{i=1}^N$ and the particle filter estimate $\{(\omega_t^i, \xi_t^i)\}_{i=1}^N$, where τ_t^i is an estimate of $t^{-1}\mathbf{T}_t s_t(\xi_t^i)$. When updating the estimates, the particle cloud is first propagated under the parameters θ_t, whereupon indices $\{J_{t+1}^{(i,j)}\}_{j=1}^{\tilde{N}}$

are drawn from $\mathsf{Pr}(\{\omega_t^\ell q_{\theta_t}(\xi_t^\ell, \xi_{t+1}^i)\}_{\ell=1}^N)$ using the accept-reject sampling approach proposed earlier. The auxiliary statistics are then updated through

$$\tau_{t+1}^i = \tilde{N}^{-1} \sum_{j=1}^{\tilde{N}} \left\{ (1 - \gamma_{t+1}) \tau_t^{J_{t+1}^{(i,j)}} + \gamma_{t+1} \tilde{s}_t(\xi_t^{J_{t+1}^{(i,j)}}, \xi_{t+1}^i) \right\} .$$

Finally the updated parameters are given by

$$\theta_{t+1} = \Lambda \left(\sum_{i=1}^N \frac{\omega_{t+1}^i}{\Omega_{t+1}} \tau_{t+1}^i \right) ,$$

where Λ is the maximization function. The algorithm is initialized by setting $\tau_0^i = 0$ and θ_0 arbitrarily.

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, 50:241–254, 2002.

[3] A. Bryson and M. Frazier. Smoothing for linear and nonlinear dynamic systems. Technical Report TDR 63-119, Aero. Sys. Div. Wrigth-Patterson Air Force Base, Ohio, 1963.

[4] O. Cappé. Ten years of HMMs (online bibliography 1989–2000), March 2001.

[5] O. Cappé. Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749, 2011.

[6] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings*, 95(5):899–924, 2007.

[7] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models.* Springer, 2005.

[8] S. Chib, F. Nadari, and N. Shephard. Markov chain Monte Carlo methods for stochastic volatility models. *J. Econometrics*, 108:281–316, 2002.

[9] P. Del Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications.* Springer, 2004.

[10] P. Del Moral, A. Doucet, and S. Singh. A Backward Particle Interpretation of Feynman-Kac Formulae. *ESAIM M2AN*, 44(5):947–975, 2010.

[11] P. Del Moral, A. Doucet, and S. Singh. Forward smoothing using sequential Monte Carlo. Technical report, Cambridge University, 2010.

[12] P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré*, 37:155–194, 2001.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39(1):1–38 (with discussion), 1977.

[14] R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential Monte Carlo smoothing for general state space hidden Markov models. *Ann. Appl. Probab.*, 21(6):2109–2145, 2011.

[15] R. Douc, E. Moulines, and J. Olsson. Long-term stability of sequential Monte Carlo methods under verifiable conditions. *Ann. Appl. Probab.*, 24(5):1767–1802, 2014.

[16] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer, New York, 2001.

[17] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.*, 10:197–208, 2000.

[18] P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010.

[19] S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *J. Am. Statist. Assoc.*, 50:438–449, 2004.

[20] N. Gordon, D. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Radar Signal Process.*, 140:107–113, 1993.

[21] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *J. Finance*, 42:281–300, 1987.

[22] M. Hürzeler and H. R. Künsch. Monte Carlo approximations for general state-space models. *J. Comput. Graph. Statist.*, 7:175–193, 1998.

[23] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[24] R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng., Trans. ASME, Series D*, 83(3):95–108, 1961.

[25] G. Kitagawa. Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 1:1–25, 1996.

[26] R. Kohn and C. F. Ansley. A fast algorithm for signal extraction, influence and cross-validation in state space models. *Biometrika*, 76:65–79, 1989.

[27] T. Koski. *Hidden Markov Models for Bioinformatics.* Kluwer, 2001.

[28] J. Olsson, O. Cappé, R. Douc, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models. *Bernoulli*, 14(1):155–179, 2008. arXiv:math.ST/0609514.

[29] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–285, February 1989.

[30] L. R. Welch. Hidden Markov models and the Baum-Welch algorithm. *IEEE Inf. Theory Soc. Newslett.*, 53(4), 2003.

[31] C. F. J. Wu. On the convergence properties of the EM algorithm. *Ann. Statist.*, 11:95–103, 1983.