BACKWARD SEQUENTIAL MONTE CARLO FOR MARGINAL SMOOTHING

Joel Kronander[1], *Thomas B. Schön*[2], *Johan Dahlin*[3]

[1] Dept. of Science and Technology, Linköping University, joel.kronander@liu.se
[2] Dept. of Information Technology, Uppsala University, thomas.schon@it.uu.se
[3] Dept. of Electrical Engineering, Linköping University, johan.dahlin@liu.se

ABSTRACT

In this paper we propose a new type of particle smoother with linear computational complexity. The smoother is based on running a sequential Monte Carlo sampler backward in time after an initial forward filtering pass. While this introduces dependencies among the backward trajectories we show through simulation studies that the new smoother can outperform existing forward-backward particle smoothers when targeting the marginal smoothing densities.

Index Terms— Sequential Monte Carlo, Particle filter, Particle smoother, Forward-backward algorithms.

1. INTRODUCTION

Consider a general state space model (SSM)

$$x_0 \sim \mu(x_0) \tag{1a}$$

$$x_t | x_{t-1} \sim f(x_t | x_{t-1}) \tag{1b}$$

$$y_t | x_t \sim g(y_t | x_t) \tag{1c}$$

where $x_t \in \mathsf{X}$ and $y_t \in \mathsf{Y}$ denote the latent state and is the observation at time t, respectively. We are interested in inferring the marginal smoothing distributions $p(x_t|y_{1:T})$, where $y_{1:k} = \{y_1, y_2, ..., y_k\}$ and T is the number of time steps. To this end, we consider forward-backward particle smoothing algorithms. These algorithms are based on running a backward smoothing pass after an initial forward particle filter, providing approximations of the filtering distributions $p(x_t|y_{1:t})$. Several variants of backward smoothers have been proposed in the literature, see e.g. Lindsten and Schön [1] for an extensive survey. Our contribution in this work is to show how the marginal smoothing distributions can be targeted using Sequential Monte Carlo (SMC) in a backward smoothing pass with linear computational complexity. Resampling operations enable high computational efficiency by focusing the computational effort on the most likely backward particle trajectories. This is in stark contrast to previous forward-backward particle smoothers which all generate independent backward trajectories.

There are several previous smoothing algorithms related to our work. The *fixed-lag smoother* [2] approximates the marginal smoothing distributions at time t, by the forward filtering particles at time $t + \Delta$, relying on the fact that $p(x_t|y_{1:T}) \approx p(x_t|y_{1:\min(t+\Delta,T)})$ for some large enough Δ. While simple to implement, the choice of the lag Δ is difficult to tune and for poorly mixing models Δ can be so large that the particle approximation degenerates. The *Forward Filter Backward Smoothing (FFBSm)* algorithm introduced by Doucet *et al.* [3] circumvents this drawback by adding a backward smoothing pass to update the particle weights from the forward filter. However, this comes at the cost of a quadratic computational complexity, $\mathcal{O}(N^2)$, where N is the number of filter particles. Godsill *et al.* [4] proposed the *Forward Filtering Backward Simulation (FFBSi)* algorithm which targets the joint smoothing density by independently sampling M new particle trajectories backwards in time. The original FFBSi algorithm exhibits a complexity of $\mathcal{O}(NM)$, but have recently been improved to linear complexity $\mathcal{O}(N)$, by considering rejection sampling [5, 6] and Markov chain Monte Carlo (MCMC) based backward simulators [7, 8].

Based on simulation studies we show that the proposed particle smoother can outperform existing forward-backward smoothers.

2. BACKWARD SMC SMOOTHER

Consider the marginal smoothing distribution at time t

$$\begin{aligned} p(x_t|y_{1:T}) &= \int p(x_t|x_{t+1}, y_{1:t}) p(x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1} \\ &= \int \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1} \\ &= \int \frac{p(x_t|y_{1:t}) g(y_{t+1}|x_{t+1}) f(x_{t+1}|x_t) p(x_{t+1}|y_{1:T})}{p(x_{t+1}|y_{1:t+1}) p(y_{t+1}|y_{1:t})} \mathrm{d}x_{t+1} \end{aligned} \tag{2}$$

This work was supported by: the Swedish Foundation for Strategic Research (SSF) through grant IIS11-0081, the project Probabilistic modeling of dynamical systems (Contract number: 621-2013-5524) funded by the Swedish Research Council and CADICS, a Linnaeus Center also funded by the Swedish Research Council.

where we used the expression

$$p(x_{t+1}|y_{1:t}) = \frac{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})}{g(y_{t+1}|x_{t+1})} \tag{3}$$

to establish the last equality. An initial forward filtering sweep generates the weighted particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ for $t = 1, \ldots, T$ providing an empirical point mass approximation of the filtering distribution

$$\widehat{p}(x_t|y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t) \tag{4}$$

where $\delta_{x_0}(x)$ denotes a Dirac delta mass located at x_0. Substituting this empirical approximation obtained from the forward filter into (2) also provides an expression for the marginal smoothing distribution,

$$\widehat{p}(x_t|y_{1:T}) = \sum_{i=1}^{N} w_t^i z_t^i \delta_{x_t^i}(x_t) \tag{5a}$$

$$z_t^i = \int \frac{g(y_{t+1}|x_{t+1})f(x_{t+1}|x_t^i)p(x_{t+1}|y_{1:T})}{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})} \mathrm{d}x_{t+1} \tag{5b}$$

To compute approximations of the marginal smoothing distribution (5), we proceed in a recursive fashion backward in time. The marginal smoothing distribution at time T can trivially be estimated from the weighted particle system, $\{x_T^i, w_T^i\}_{i=1}^N$, targeting the filtering distribution at time T. To compute the marginal smoothing distributions for $t = T-1, \ldots, 1$ we assume that there exists a weighted particle system $\{\widetilde{x}_{t+1}^j, \widetilde{w}_{t+1}^j\}_{j=1}^M$ targeting $p(x_{t+1}|y_{1:T})$. For $t = T-1$ this system can be obtained by resampling M particles from the N filter particles at time T. By substituting $\{\widetilde{x}_{t+1}^j, \widetilde{w}_{t+1}^j\}_{j=1}^M$ for $p(x_{t+1}|y_{1:T})$ in the expression for the marginal smoothing distribution (5) we obtain

$$\widehat{p}(x_t|y_{1:T}) \propto \sum_{i=1}^{N}\sum_{j=1}^{M} w_t^i \widetilde{w}_{t+1}^j \frac{g(y_{t+1}|\widetilde{x}_{t+1}^j)f(\widetilde{x}_{t+1}^j|x_t^i)}{p(\widetilde{x}_{t+1}^j|y_{1:t+1})}\delta_{x_t^i}(x_t) \tag{6}$$

Assuming that the particle system $\{\widetilde{x}_{t+1}^j, \widetilde{w}_{t+1}^j\}_{j=1}^M$ at time $t+1$ is in the support of the particle system obtained in the forward filtering pass $\{x_{t+1}^i, w_{t+1}^i\}_{i=1}^N$ we can use the empirical approximation provided by that particle system as an approximation for $p(\widetilde{x}_{t+1}^j|y_{1:t+1})$, thus

$$\widehat{p}(\widetilde{x}_{t+1}^j|y_{1:t+1}) = \begin{cases} w_{t+1}^i & \text{if } \widetilde{x}_{t+1}^j = x_{t+1}^i \\ 0 & \text{if } \widetilde{x}_{t+1}^j \neq x_{t+1}^i \end{cases} \tag{7}$$

We propose to target the approximation of the marginal smoothing densities given by plugging in (7) into (6) using importance sampling. To this end we consider an approach in the same spirit as the auxiliary particle filter [9].

Algorithm 1: Backward SMC smoother

Input : Forward filtering particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ for $t = 1, \ldots, T$.
Output: Backward particle systems $\{\widetilde{x}_t^j, \widetilde{w}_t^j\}_{j=1}^M$ for $t = 1, \ldots, T$.

Sample $\{b_T^m\}_{m=1}^M \sim Cat\left(\{w_T^i\}_{i=1}^N\right)$
Set $\widetilde{X}_T^m = X_T^{b_T^m}$ for $m = 1, \ldots, M$
for $t = T-1, \ldots, 1$ **do**
    for $j = 1, \ldots, M$ **do**
        Sample $a_t^j \sim Cat\left(\{w_t^i\}_{i=1}^N\right)$
        Sample $b_t^j \sim Cat\left(\{\frac{\widetilde{w}_{t+1}^k g(y_{t+1}|\widetilde{x}_{t+1}^{k_t})}{\widehat{p}(\widetilde{x}_{t+1}^{k_t}|y_{1:t+1})}\}_{k=1}^M\right)$
        Set $\widetilde{x}_t^j = x_t^{a_t^j}$
        Set $\widetilde{w}_t^j = f(\widetilde{X}_{t+1}^{b_t^j}|\widetilde{X}_t^j)$
    end
    Set $\widetilde{w}_t^j = \frac{\widetilde{w}_t^j}{\sum_{j=1}^M \widetilde{w}_t^j}$ for $j = 1, \ldots, M$
end

Let us introduce two auxiliary uniformly distributed random variables a_t and b_t, which are defined on the index sets $\{1, \ldots, N\}$ and $\{1, \ldots, M\}$, respectively. Let π_t be a probability density defined on $\mathsf{X} \times \{1, \ldots, N\} \times \{1, \ldots, M\}$ according to

$$\pi_t(x_t, a_t, b_t) \propto w_t^{a_t}\widetilde{w}_{t+1}^{b_t}\frac{g(y_{t+1}|\widetilde{x}_{t+1}^{b_t})f(\widetilde{x}_{t+1}^{b_t}|x_t^{a_t})}{\widehat{p}(\widetilde{x}_{t+1}^{b_t}|y_{1:t+1})}\delta_{x_t^{a_t}}(x_t) \tag{8}$$

Marginalizing this distribution over a_t and b_t gives the expression for the marginal smoothing distribution given in (6). Targeting (8) with M samples from the proposal distribution $q(x_t, a_t, b_t) = w_t^{a_t} q(b_t|x^{a_t})$ gives a weighted particle system $\{\widetilde{x}_t^j, \widetilde{w}_t^j\}_{j=1}^M$ targeting (6) with weights

$$\widetilde{w}_t^j \propto \frac{\widetilde{w}_{t+1}^{b_t} g(y_{t+1}|\widetilde{x}_{t+1}^{b_t})f(\widetilde{x}_{t+1}^{b_t}|x_t^{a_t})}{\widehat{p}(\widetilde{x}_{t+1}^{b_t}|y_{1:t+1})q(b_t|x^{a_t})} \tag{9}$$

The optimal proposal distribution for b_t in the sense of minimal variance is $q(b_t|x^{a_t}) = \frac{\widetilde{w}_{t+1}^{b_t} g(y_{t+1}|\widetilde{x}_{t+1}^{b_t})f(\widetilde{x}_{t+1}^{b_t}|x_t^{a_t})}{\widehat{p}(\widetilde{x}_{t+1}^{b_t}|y_{1:t+1})}$. However, to sample from this proposal for each sampled index a_t^j results in an algorithm with complexity $\mathcal{O}(NM)$. If we instead sample b_t independently from a_t using e.g. $q(b_t) = \frac{\widetilde{w}_{t+1}^{b_t} g(y_{t+1}|\widetilde{x}_{t+1}^{b_t})}{\widehat{p}(\widetilde{x}_{t+1}^{b_t}|y_{1:t+1})}$, the computational complexity is reduced to $O(N+M)$. For a bootstrap particle filter [10] this choice leads to a particularly simple proposal distribution $q(b_t) = \widetilde{w}_{t+1}^{b_t}$ as $w_{t+1}^i = g(y_{t+1}|x_{t+1}^i)$.

The variance can also be reduced by considering stratification in the auxiliary index spaces [11], commonly referred to as *stratified resampling*. The above development is summarized in Algorithm 1, where $Cat(\{p^i\}_{i=1}^n)$ denotes the categorical distribution on $\{1, \ldots, N\}$ with probabilities $\{p^i\}_{i=1}^N$.

3. NUMERICAL ILLUSTRATION

In this section the proposed methods are compared to the fixed-lag smoother [2] (FS) using a lag of 5, FFBSi [4], a recent version of RS-FFBSi [6] with a fixed number of rejections sampling tests before an exhaustive evaluation of the smoothing weights ($R_{\max} = M/5$ was used to generate the results presented here) and the MH-FFBSi method [8] using $K = 10$ MCMC step for each particle update. All smoothers are based on a standard particle filter (PF) sampling from the state space dynamics $f(x_t|x_{t-1})$ and resampling when the effective sample size (ESS) is less than $\frac{2}{3}N$. The reported runtimes were generated using MATLAB and includes the runtime of both the forward filter and the backward smoother.

3.1. Linear Gaussian SSM

Consider a generic ten dimensional linear Gaussian SSM

$$x_{t+1} = Ax_t + w_t, \qquad w_t \sim \mathcal{N}(0, Q) \tag{10a}$$
$$y_t = Cx_t + v_t, \qquad v_t \sim \mathcal{N}(0, R) \tag{10b}$$

For this model 50 independent realizations of A and C were generated using the MATLAB command `drss(10,10,0)`, and Q and R were set to the identity matrix. For each model realization, 10 datasets each of length $T = 100$ were considered. Table 1 show runtimes and average mean squared error (MSE) compared to the Rauch-Tung-Striebel smoother [12] solution for $N = 200$ and $M = 100$. For this model the fixed-lag smoother performs worse than the other smoothers. Due to the high dimensionality of the model, the rejection sampling based ARS-FFBSi [6] smoother performs on par with the FFBSi smoother in terms of efficiency. The MH-FFBSi algorithm is approximately five times as fast as the FFBSi smoother, yielding similar MSE. The proposed smoother performs similarly to previous forward-backward smoothers in terms of MSE, but at a lower computational cost. For these parameter settings about seventeen times faster than the FFBSi algorithm and about four times faster than MH-FFBSi.

3.2. Nonlinear model

Consider the nonlinear radar-type tracking model given in [8]

$$\begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} I_2 & I_2 \\ 0_2 & I_2 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} + w_t \tag{11a}$$

$$\begin{bmatrix} b_t \\ r_t \end{bmatrix} = \begin{bmatrix} \arctan(\frac{y_t}{x_t}) \\ \sqrt{x_t^2 + y_t^2} \end{bmatrix} + v_t \tag{11b}$$

where I_2 and 0_2 denote the 2×2 identity matrix and 2×2 zero matrix, respectively. The random variables w_t and v_t are independent and zero-mean Gaussian distributed with covariance matrices Q and R respectively

$$Q = \begin{bmatrix} \frac{1}{3}I_2 & \frac{1}{2}I_2 \\ \frac{1}{2}I_2 & I_2 \end{bmatrix} \qquad R = \begin{bmatrix} \sigma_B^2 & 0 \\ 0 & \sigma_R^2 \end{bmatrix}$$

where $\sigma_B^2 = \left(\frac{\pi}{720}\right)^2$ and $\sigma_R^2 = 0.1$. For this model we averaged the results over 100 data sets where each dataset was generated with $T = 300$. Table 2 presents the average MSE and runtimes for different settings of N and M. For the same number of particles in the forward and backward sweep the MH-FFBSi algorithm performs better than the proposed smoother, but at a higher computational cost. When using twice as many particles ($N = 2000, M = 400$) the proposed smoother gives the same MSE with approximately half the runtime compared to the MH-FFBSi algorithm with ($N = 1000, M = 200$) particles.

4. DISCUSSION

Computational complexity - The proposed algorithm in general requires less computations per sampled backward trajectory than previous smoothers. Apart from the cost of sampling from the categorical distributions, a maximum of two function evaluations have to be performed for each particle at each time step. When the forward filter corresponds to a bootstrap particle filter the proposed algorithm is even more efficient, only requiring one function evaluation per particle per time step. In contrast, the recently proposed MH-FFBSi algorithm, always performs K iterations per particle per time step, each of which entails one new function evaluation, where K in the range of $3 - 20$ often yields good results in practice. The proposed smoother thus requires roughly $(K - 2)MT$ less functions evaluations then the MH-FFBSi algorithm for the same number of backward particles. However, a drawback with the proposed smoother is that the resampling of the backward trajectories can limit the efficiency of a parallel implementation, similar to forward SMC filtering, while for other particle smoothers the backward pass is often straightforward to parallelise.

Empirical Convergence - In numerical experiments we have found that the proposed SMC smoother exhibits a non-vanishing bias when increasing the number of forward and backward particles. The bias tend to be in the same order as for the MH-FFBSi algorithm with $K \approx 3-5$. To remove any remaining bias for a large number of particles, the MCMC smoother proposed by Dubarry and Douc [7] could be used in a post process step. To investigate other methods to remove the bias is an interesting venue for future work.

Future work - Recent forward-backward particle smoothers have enabled new particles to be sampled in the backward pass, not in the support of the forward filter approximations.

	FS [2]	**FFBSm** [3]	**FFBSi** [4]	**ARS-FFBSi** [6]	**MH-FFBSi** [8]	**Algorithm 1**
MSE	0.75	0.66	0.66	0.66	0.66	0.66
Runtime	0.06	6.90	2.02	1.75	0.38	0.10

Table 1. Average MSE and runtime for the ten dimensional linear Gaussian model (10), $N = 200, M = 100$

	MH-FFBSi [8] N = 1000 M = 200	**MH-FFBSi** [8] N = 2000 M = 400	**MH-FFBSi** [8] N = 4000 M = 800	**Algorithm 1** N = 1000 M = 200	**Algorithm 1** N = 2000 M = 400	**Algorithm 1** N = 4000 M = 800
MSE	2.8	2.4	2.3	3.4	2.8	2.5
Runtime	1.15	1.43	2.02	0.39	0.53	0.85

Table 2. Average MSE and runtimes for the nonlinear tracking model (11). Columns are sorted according to MSE.

One way to extend the proposed smoother to sample new particle positions in the backward pass is to approximate the filtering distribution at time t in the expression for the marginal smoothing distribution (2) with the propagated empirical filtering distribution from $t-1$, similar to [8]. To approximate the denominator in the backward kernel, the approximation, $p(\widetilde{x}_{t+1}^j|y_{1:t}) = \sum_{k=1}^N w_t^k f(\widetilde{x}_{t+1}^j|x_t^k)\delta_{x_t^k}(x_t)$, can be used. However, this results in an algorithm with quadratic computational complexity. To derive a backward proposing SMC smoother of linear complexity it would be interesting to investigate methods to approximate $p(\widetilde{x}_{t+1}^j|y_{1:t+1})$ using for example kernel density estimation based on the forward particle system.

Many extensions similar to those developed for forward SMC filters could also be considered for the proposed backward SMC sampler. For example resample-move [13] algorithms, adaptive resampling and parallel resampling [14]. Investigating such extensions in the context of backward simulation presents several interesting venues for future work.

5. REFERENCES

[1] Fredrik Lindsten and Thomas B. Schön, "Backward simulation methods for Monte Carlo statistical inference," *Foundations and Trends in Machine Learning*, vol. 6, no. 1, pp. 1–143, 2013.

[2] Genshiro Kitagawa and Seisho Sato, *Sequential Monte Carlo methods in practice*, chapter Monte Carlo smoothing and self-organizing state-space model, pp. 177–195, Springer-Verlag, 2001.

[3] Arnaud Doucet, Simon Godsill, and Christophe Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[4] Simon J. Godsill, Arnaud Doucet, and Mike West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

[5] Randal Douc, Aurélien Garivier, Eric Moulines, and Jimmy Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *The Annals of Applied Probability*, vol. 21, no. 6, pp. 2109–2145, 2011.

[6] Ehsan Taghavi, Fredrik Lindsten, Lennart Svensson, and Thomas B. Schön, "Adaptive stopping for fast particle smoothing," in *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, number 4.

[7] Cyrille Dubarry and Randal Douc, "Particle approximation improvement of the joint smoothing distribution with on-the-fly variance estimation," *arXiv preprint arXiv:1107.5524*, 2011.

[8] Pete Bunch and Simon Godsill, "Improved particle approximations to the joint smoothing distribution using Markov Chain Monte Carlo," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 956–963, 2013.

[9] Michael K. Pitt and Neil Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[10] Neil J. Gordon, David J. Salmond, and Adrian F.M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, 1993, vol. 140, pp. 107–113.

[11] James Carpenter, Peter Clifford, and Paul Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[12] H. E. Rauch, C.T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, 1965.

[13] Walter R. Gilks and Carlo Berzuini, "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[14] Pierre E. Jacob Lawrence M. Murray, Anthony Lee, "Rethinking resampling in the particle filter on graphics processing units," *arXiv:1301.4019 [stat.CO]*, 2013.