A Lagged Particle Filter for Stable Filtering of certain High-Dimensional State-Space Models

Hamza Ruzayqat, Almud Eerain, Alexandros Beskos, Dan Crisan, Ajay Jasra and Nikolas Kantas

1King Abdullah University of Science and Technology, 2University College London, 3Imperial College London

hamza.ruzayqat@kaust.edu.sa

Abstract

We consider the problem of high-dimensional filtering of state-space models (SSMs) at discrete times using sequential Monte Carlo (SMC). The main issue is that as the dimension of the hidden state increases, one may need the number of SMC samples to be $N \sim \exp(n)$ for some $n > 1$. We develop an SMC method to recursively estimate expectations with respect to approximations of the smoothing distributions and prove for a class of SSMs that the bias of our approximation is uniformly controlled in $n$ and exponentially small in time and that as $d \to \infty$ the cost to achieve a stable mean square error in estimation, for classes of expectations, is of $O(Nn^p)$ per-unit time. The proposed method is showed to be superior to commonly used methods such as the ensemble Kalman filter (EnKF), ensemble transform Kalman filter (ETKF).

1. State Space Model

We have two sequences of random variables $(Y_{1:n})_{n \in \mathbb{N}}$, $(X_{1:n})_{n \in \mathbb{N}}$ such that $Y_{1:n} \subset \mathbb{R}^n$, $n \in \mathbb{N}$, and $X_{1:n} \subset \mathbb{R}^n$, $n \in \mathbb{N}$. We endow $\mathbb{R}^n$ with an associated $\sigma$-field $\mathbb{F}$. Consider the state-space model (e.g. [1]), where for $n \geq 1$: $X_{1:n} = \mathcal{A}[X_{1:n-1}, Y_{1:n}] = \int f(x_{n-1}, x_n) \nu_{n-1} \, dx_{n-1}$, and $P(Y_{1:n} | X_{1:n}) = \int g(x_n, y_n) \nu_n \, dx_n$.

The initial condition $X_1 = x_1$ is K-assumed given and fixed, $\nu_n(x_n)$ is a positive probability density w.r.t. the $\sigma$-finite measure $\nu_n$ for each $x_n \in \mathbb{R}^n$. $\nu_n$ is a positive probability density w.r.t. the $\sigma$-finite measure $\mathbb{F}$ for each $x \in \mathbb{R}^n$. For $n \in \mathbb{N}$ we define the smoothing density:}

$$
\rho_n(x_1) \propto \prod_{k=1}^{n} \int f(x_{k-1}, x_k) \nu_k \, dx_k,
$$

(1)

where $\nu_k = (\nu_k(x_1))_{x_1}$ are fixed and known observations. The objective is to recursively estimate the so-called filter for each $n \in \mathbb{N}$:

$$
\hat{\pi}_n(x_n) = \frac{\rho_n(x_n)}{\sum_{x_1} \rho_n(x_1)}
$$

2. Lagged Approximation of Smoothing Distribution

Given $L \in \mathbb{N}$. From time $n = 1, \ldots, L$, set $\hat{\pi}_{n,L}(x_n) = \hat{\pi}_n(x_n)$ and use the SMC sampler method [2] to sample from $\rho_L$, where the whole path $x_{1:L}$ is updated in the Markov chain Monte Carlo (MCMC) step of the SMC sampler. Next, for $L = n + 1, \ldots, L + 1$, we introduce the lagged approximation

$$
\hat{\pi}_{n+1,L}(x_{n+1}) \propto \prod_{k=1}^{n+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k \prod_{k=n+2}^{L+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k
$$

where $\mu(x) = f(x_{n+1})$ and $\pi_{n+1}(x_{n+1})$, $p = 1, \ldots, r$, is a sequence of user-specified probability densities on $X$ used to create some kind of independence between the states. The sequential importance sampling incremental weights only depend on $x_{n+1}$, thereby forgetting earlier positions of $x_{n+1}$. Note under the choice $\mu(x_{n+1}) = \prod_{k=n+2}^{L+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k$ one goes back to (1).

3. SMC Sampler Algorithm to Sample from $\hat{\pi}_n$

Part 1: Let $n \in \{1, \ldots, L\}$. Given the $N$ paths $(x_{1:n})_{n \in \mathbb{N}}$ with weights $(\pi_{n:n})_{n \in \mathbb{N}}$, we sample $\hat{\pi}_n(x_{1:n})$ from the proposal $g(x_{1:n}) = \frac{\rho_n(x_{1:n})}{\sum_{x_1} \rho_n(x_1)}$. Then we sequentially update using the SMC sampler [2], from the

$$
\hat{\pi}_n(x_{1:n}) \propto \prod_{k=1}^{n} \int f(x_{n-1}, x_n) \nu_n \, dx_n \prod_{k=n+2}^{L+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k
$$

starting at $\hat{\pi}_1$ and ending at $\hat{\pi}_L = \pi_L$, for some $L \in \mathbb{N}$. Part 2: Now let $n \geq L + 1$. Given $N$ paths $(x_{1:n})_{n \in \mathbb{N}}$ with weights $(\pi_{n:n})_{n \in \mathbb{N}}$. We sample the particles $x_{1:n}^1, \ldots, x_{1:n}^N$ from the proposal $g(x_{1:n}) = \prod_{k=1}^{n} \int f(x_{n-1}, x_n) \nu_n \, dx_n$. Setting $\rho_n(x) = f(x_{n+1})$, we will sample sequentially from the distributions

$$
\hat{\pi}_n(x_{1:n}) \propto \prod_{k=1}^{n} \int f(x_{n-1}, x_n) \nu_n \, dx_n \prod_{k=n+2}^{L+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k
$$

starting at $\hat{\pi}_1$ and ending at $\hat{\pi}_L = \pi_L$, for some $L \in \mathbb{N}$. The numbers $0 = \phi_0 < \cdots < \phi_n = 1$ are called temperatures. In both Part 1 and Part 2, for $L = 1, \ldots, \hat{\pi}_L$, inside the SMC sampler algorithm, one computes the weights for each particle $i = 1, \ldots, N$ as

$$
W^{(i)}_n = \frac{\pi_{n:n}(x_{1:n}^{(i)})}{\hat{\pi}_n(x_{1:n}^{(i)})} g(x_{1:n}^{(i)}) \prod_{k=n+2}^{L+1} \int f(x_k, x_{k+1}) \nu_k \, dx_k
$$

then runs an adaptive resampling scheme to resample the $N$ paths and finally runs an MCMC step with $N$ iterations and $\hat{\pi}_n(x_{1:n})$ the invariant distribution of the MCMC kernel, where the whole path $x_{1:n}$ gets updated in Part 1 and only $x_{n+1}$ gets updated in Part 2.

4. Numerical Results

Let $X_n \in \mathbb{R}^2$, for $n = 1, \ldots, T$ and $Y_n \in \mathbb{R}^2$, for $m = 1, \ldots, M$, for some $T, M \in \mathbb{N}$. Assume that $X_n = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^2 \times \cdots \times \mathbb{R}^2$. We consider the following SSM:

$\nonumber X_n = g(x_{n-1}^{(1)}, X_{n-1}^{(2)}, \ldots, X_{n-1}^{(N)})$, $n = 1, \ldots, T$, $g : \mathbb{R}^2 \times \cdots \times \mathbb{R}^2 \to \mathbb{R}^2$, $R_n = \mathbb{R}^2$, $R_n = \mathbb{R}^2$, $W_n = \mathcal{N}(0, I_2)$.

Examples: (1) Linear Model $g(x_{n-1}^{(1)}, X_{n-1}^{(2)}, \ldots, X_{n-1}^{(N)})$ is the 4th-order Runge-Kutta approximation of the Lorenz 96 system and (3) Shallow Water Model $g(x_{n-1}^{(1)}, X_{n-1}^{(2)}, \ldots, X_{n-1}^{(N)})$ is the finite volume (FV) solution of the conservative SW equations.

Acknowledgements

AJ & HR were supported by KAUST baseline funding. The work of DC has been partially supported by European Research Council (ERC) Synergy grant STUDD-DLV-6564. NK was supported by a J.P. Morgan AI Research Award.

References