Sequential Monte Carlo Methods

Nando De Freitas & Arnaud Doucet
UBC

State-Space Models

- $\{X_k\}_{k\geq 1}$ hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(x_1) \text{ and } X_k | (X_{k-1} = x_{k-1}) \sim f(x_k | x_{k-1}).$$

State-Space Models

- $\{X_k\}_{k\geq 1}$ hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu(x_1) \text{ and } X_k | (X_{k-1} = x_{k-1}) \sim f(x_k | x_{k-1}).$$

- $\{Y_k\}_{k\geq 1}$ observed $\mathcal{Y}$-valued process with observations conditionally independent given $\{X_k\}_{k\geq 1}$ with

$$Y_k | (X_k = x_k) \sim g(y_k | x_k).$$

State-Space Models

- $\{X_k\}_{k\geq 1}$ hidden $\mathcal{X}$-valued Markov process with

$$X_1 \sim \mu\left(x_1\right) \text{ and } X_k \middle| \left(X_{k-1} = x_{k-1}\right) \sim f\left(x_k \middle| x_{k-1}\right).$$

- $\{Y_k\}_{k\geq 1}$ observed $\mathcal{Y}$-valued process with observations conditionally independent given $\{X_k\}_{k\geq 1}$ with

$$Y_k \middle| \left(X_k = x_k\right) \sim g\left(y_k \middle| x_k\right).$$

- **Main Objective**: Estimate $\{X_k\}_{k\geq 1}$ given $\{Y_k\}_{k\geq 1}$ online/offline.

Inference in State-Space Models

- Given observations $y_{1:n} := (y_1, y_2, \ldots, y_n)$, inference about $X_{1:n} := (X_1, ..., X_n)$ relies on the posterior

$$p\left(x_{1:n} \middle| y_{1:n}\right) = \frac{p\left(x_{1:n}, y_{1:n}\right)}{p\left(y_{1:n}\right)}$$

where

$$p\left(x_{1:n}, y_{1:n}\right) = \underbrace{\mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \middle| x_{k-1}\right)}_{p(x_{1:n})} \underbrace{\prod_{k=1}^{n} g\left(y_k \middle| x_k\right)}_{p(\,y_{1:n}|x_{1:n})},$$

$$p\left(y_{1:n}\right) = \int \cdots \int p\left(x_{1:n}, y_{1:n}\right) dx_{1:n}$$

Inference in State-Space Models

- Given observations $y_{1:n} := (y_1, y_2, \ldots, y_n)$, inference about $X_{1:n} := (X_1, ..., X_n)$ relies on the posterior

$$p(x_{1:n} | y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})}$$

where

$$p(x_{1:n}, y_{1:n}) = \underbrace{\mu(x_1) \prod_{k=2}^{n} f(x_k | x_{k-1})}_{p(x_{1:n})} \underbrace{\prod_{k=1}^{n} g(y_k | x_k)}_{p(y_{1:n} | x_{1:n})},$$

$$p(y_{1:n}) = \int \cdots \int p(x_{1:n}, y_{1:n}) \, dx_{1:n}$$

- We want to compute $p(x_{1:n} | y_{1:n})$ and $p(y_{1:n})$ *sequentially* in time n.

Inference in State-Space Models

- Given observations $y_{1:n} := (y_1, y_2, \ldots, y_n)$, inference about $X_{1:n} := (X_1, ..., X_n)$ relies on the posterior

$$p(x_{1:n} | y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})}$$

where

$$p(x_{1:n}, y_{1:n}) = \underbrace{\mu(x_1) \prod_{k=2}^{n} f(x_k | x_{k-1})}_{p(x_{1:n})} \underbrace{\prod_{k=1}^{n} g(y_k | x_k)}_{p(y_{1:n} | x_{1:n})},$$

$$p(y_{1:n}) = \int \cdots \int p(x_{1:n}, y_{1:n}) \, dx_{1:n}$$

- We want to compute $p(x_{1:n} | y_{1:n})$ and $p(y_{1:n})$ *sequentially* in time n.
- For non-linear non-Gaussian models, numerical approximations are required.

Monte Carlo Methods

- Assume you can generate $X_{1:n}^{(i)} \sim p\left(x_{1:n} \middle| y_{1:n}\right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$$

Monte Carlo Methods

- Assume you can generate $X_{1:n}^{(i)} \sim p\left(x_{1:n} \middle| y_{1:n} \right)$ where $i = 1, \ldots, N$ then MC approximation is

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}} \left(x_{1:n} \right)$$

- **Integration is straightforward**

$$\int \varphi_n \left(x_{1:n} \right) \widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) dx_{1:n} = \frac{1}{N} \sum_{i=1}^{N} \varphi_n \left(X_{1:n}^{(i)} \right).$$

Monte Carlo Methods

- Assume you can generate $X_{1:n}^{(i)} \sim p\left(x_{1:n} \middle| y_{1:n} \right)$ where $i = 1, \ldots, N$ then MC approximation is

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}} \left(x_{1:n} \right)$$

- **Integration is straightforward**

$$\int \varphi_n \left(x_{1:n} \right) \widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) dx_{1:n} = \frac{1}{N} \sum_{i=1}^{N} \varphi_n \left(X_{1:n}^{(i)} \right).$$

- **Marginalisation is straightforward**

$$\widehat{p}\left(x_k \middle| y_{1:n} \right) = \int \widehat{p}\left(x_k \middle| y_{1:n} \right) dx_{1:k-1} dx_{k+1:n} = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}} \left(x_k \right)$$

Monte Carlo Methods

- Assume you can generate $X_{1:n}^{(i)} \sim p\left(x_{1:n} \middle| y_{1:n} \right)$ where $i = 1, ..., N$ then MC approximation is

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}} \left(x_{1:n} \right)$$

- **Integration is straightforward**

$$\int \varphi_n \left(x_{1:n} \right) \widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) dx_{1:n} = \frac{1}{N} \sum_{i=1}^{N} \varphi_n \left(X_{1:n}^{(i)} \right).$$

- **Marginalisation is straightforward**

$$\widehat{p}\left(x_k \middle| y_{1:n} \right) = \int \widehat{p}\left(x_k \middle| y_{1:n} \right) dx_{1:k-1} dx_{k+1:n} = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}} \left(x_k \right)$$

- **Problem**: Sampling from $p\left(x_{1:n} \middle| y_{1:n} \right)$ is impossible in general cases.

Basics of Sequential Monte Carlo Methods

- **Divide and conquer strategy**: Break the problem of sampling from $p\left(x_{1:n}\middle|\,y_{1:n}\right)$ into a collection of simpler subproblems. First approximate $p\left(x_1\middle|\,y_1\right)$ at time 1, then $p\left(x_{1:2}\middle|\,y_{1:2}\right)$ at time 2 and so on.

Basics of Sequential Monte Carlo Methods

- **Divide and conquer strategy**: Break the problem of sampling from $p\left(x_{1:n}\middle|\, y_{1:n}\right)$ into a collection of simpler subproblems. First approximate $p\left(x_1\middle|\, y_1\right)$ at time 1, then $p\left(x_{1:2}\middle|\, y_{1:2}\right)$ at time 2 and so on.
- Each target distribution is approximated by a cloud of random samples termed *particles* evolving according to *importance sampling* and *resampling* steps.

Importance Sampling

- Assume you have at time $n-1$

$$\widehat{p}\left(x_{1:n-1}\middle|\, y_{1:n-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:n-1}^{(i)}}\left(x_{1:n-1}\right).$$

Importance Sampling

- Assume you have at time $n-1$

$$\widehat{p}\left(x_{1:n-1} \middle| y_{1:n-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n-1}^{(i)}}\left(x_{1:n-1}\right).$$

- By sampling $\widetilde{X}_n^{(i)} \sim f\left(x_n \middle| X_{n-1}^{(i)}\right)$ and setting $\widetilde{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ then

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right).$$

Importance Sampling

- Assume you have at time $n-1$

$$\widehat{p}\left(x_{1:n-1}\middle| y_{1:n-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:n-1}^{(i)}}\left(x_{1:n-1}\right).$$

- By sampling $\widetilde{X}_n^{(i)}\sim f\left(x_n\middle| X_{n-1}^{(i)}\right)$ and setting $\widetilde{X}_{1:n}^{(i)}=\left(X_{1:n-1}^{(i)},\widetilde{X}_n^{(i)}\right)$ then

$$\widehat{p}\left(x_{1:n}\middle| y_{1:n-1}\right)=\frac{1}{N}\sum_{i=1}^{N}\delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right).$$

- Our target at time n is

$$p\left(x_{1:n}\middle| y_{1:n}\right)=\frac{g\left(y_n\middle| x_n\right)p\left(x_{1:n}\middle| y_{1:n-1}\right)}{\int g\left(y_n\middle| x_n\right)p\left(x_{1:n}\middle| y_{1:n-1}\right)dx_n}$$

so by substituting $\widehat{p}\left(x_{1:n}\middle| y_{1:n-1}\right)$ to $p\left(x_{1:n}\middle| y_{1:n-1}\right)$ we obtain

$$\widetilde{p}\left(x_{1:n}\middle| y_{1:n}\right)=\sum_{i=1}^{N}W_n^{(i)}\delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right),\quad W_n^{(i)}\propto g\left(y_n\middle|\widetilde{X}_{1:n}^{(i)}\right).$$

Resampling

- We have a "weighted" approximation $\widetilde{p}\left(x_{1:n} \mid y_{1:n}\right)$ of $p\left(x_{1:n} \mid y_{1:n}\right)$

$$\widetilde{p}\left(x_{1:n} \mid y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right).$$

Resampling

- We have a "weighted" approximation $\widetilde{p}\left(x_{1:n}|\,y_{1:n}\right)$ of $p\left(x_{1:n}|\,y_{1:n}\right)$

$$\widetilde{p}\left(x_{1:n}|\,y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right).$$

- To obtain N samples $X_{1:n}^{(i)}$ approximately distributed according to $p\left(x_{1:n}|\,y_{1:n}\right)$, we just resample

$$X_{1:n}^{(i)} \sim \widetilde{p}\left(x_{1:n}|\,y_{1:n}\right)$$

to obtain

$$\widehat{p}\left(x_{1:n}|\,y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

Resampling

- We have a "weighted" approximation $\widetilde{p}\left(x_{1:n} \middle| y_{1:n} \right)$ of $p\left(x_{1:n} \middle| y_{1:n} \right)$

$$\widetilde{p}\left(x_{1:n} \middle| y_{1:n} \right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n} \right).$$

- To obtain N samples $X_{1:n}^{(i)}$ approximately distributed according to $p\left(x_{1:n} \middle| y_{1:n} \right)$, we just resample

$$X_{1:n}^{(i)} \sim \widetilde{p}\left(x_{1:n} \middle| y_{1:n} \right)$$

  to obtain

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n} \right).$$

- Particles with high weights are copied multiples times, particles with low weights die.

Bootstrap Filter (Gordon, Salmond & Smith, 1993)

At time $n = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)} \right).$$

Bootstrap Filter (Gordon, Salmond & Smith, 1993)

At time $n = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \; W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)} \right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

Bootstrap Filter (Gordon, Salmond & Smith, 1993)

At time $n = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)} \right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

Bootstrap Filter (Gordon, Salmond & Smith, 1993)

At time $n = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \; W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

At time $n \geq 2$

- Sample $\widetilde{X}_n^{(i)} \sim f\left(x_n | X_{n-1}^{(i)}\right)$, set $\widetilde{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ and

$$\widetilde{p}(x_{1:n} | y_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}(x_{1:n}), \; W_n^{(i)} \propto g\left(y_n | \widetilde{X}_n^{(i)}\right).$$

Bootstrap Filter (Gordon, Salmond & Smith, 1993)

At time $n = 1$

- Sample $\widetilde{X}_1^{(i)} \sim \mu(x_1)$ then

$$\widetilde{p}(x_1 | y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\widetilde{X}_1^{(i)}}(x_1), \quad W_1^{(i)} \propto g\left(y_1 | \widetilde{X}_1^{(i)}\right).$$

- Resample $X_1^{(i)} \sim \widetilde{p}(x_1 | y_1)$ to obtain $\widehat{p}(x_1 | y_1) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$.

At time $n \geq 2$

- Sample $\widetilde{X}_n^{(i)} \sim f\left(x_n | X_{n-1}^{(i)}\right)$, set $\widetilde{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ and

$$\widetilde{p}(x_{1:n} | y_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}(x_{1:n}), \quad W_n^{(i)} \propto g\left(y_n | \widetilde{X}_n^{(i)}\right).$$

- Resample $X_{1:n}^{(i)} \sim \widetilde{p}(x_{1:n} | y_{1:n})$ to obtain $\widehat{p}(x_{1:n} | y_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}(x_{1:n})$.

SMC Output

- At time n, we get

$$\widehat{p}\left(x_{1:n} \mid y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

SMC Output

- At time n, we get

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:n}\right) = \prod_{k=1}^{n} \widehat{p}\left(y_k \middle| y_{1:k-1}\right) = \prod_{k=1}^{n} \left(\frac{1}{N} \sum_{i=1}^{N} g\left(y_k \middle| \widetilde{X}_k^{(i)}\right)\right).$$

SMC Output

- At time n, we get

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:n}\right) = \prod_{k=1}^{n}\widehat{p}\left(y_k \middle| y_{1:k-1}\right) = \prod_{k=1}^{n}\left(\frac{1}{N}\sum_{i=1}^{N} g\left(y_k \middle| \widetilde{X}_k^{(i)}\right)\right).$$

- Computational complexity is $\mathcal{O}(N)$ and memory requirements $\mathcal{O}(nN)$.

SMC Output

- At time n, we get

$$\widehat{p}\left(x_{1:n}\middle|\, y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

- The marginal likelihood estimate is given by

$$\widehat{p}\left(y_{1:n}\right) = \prod_{k=1}^{n} \widehat{p}\left(y_k\middle|\, y_{1:k-1}\right) = \prod_{k=1}^{n}\left(\frac{1}{N}\sum_{i=1}^{N} g\left(y_k\middle|\, \widetilde{X}_k^{(i)}\right)\right).$$

- Computational complexity is $\mathcal{O}(N)$ and memory requirements $\mathcal{O}(nN)$.
- If we are only interested in $p\left(x_n\middle|\, y_{1:n}\right)$ or $p\left(s_n\left(x_{1:n}\right)\middle|\, y_{1:n}\right)$ where $s_n\left(x_{1:n}\right) = \Psi_n\left(x_n, s_{n-1}\left(x_{1:n-1}\right)\right)$ is fixed-dimensional then memory requirements $\mathcal{O}(N)$.

SMC on Path-Space - figures by Olivier Cappė

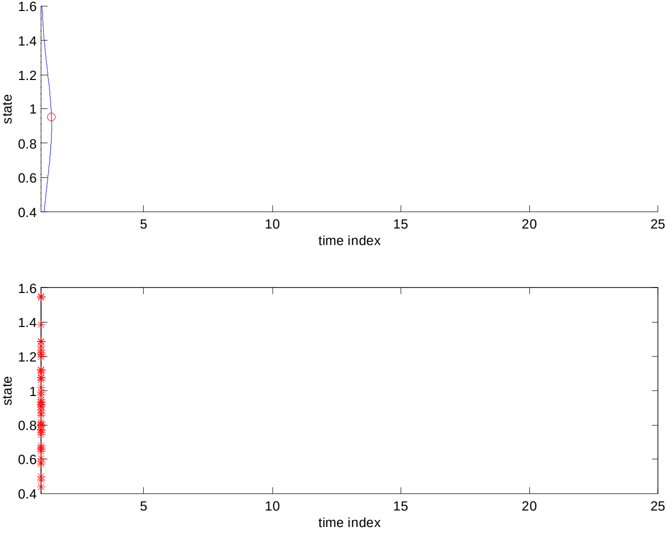


Figure: $p(x_1|y_1)$ and $\widehat{\mathbb{E}}[X_1|y_1]$ (top) and particle approximation of $p(x_1|y_1)$ (bottom)

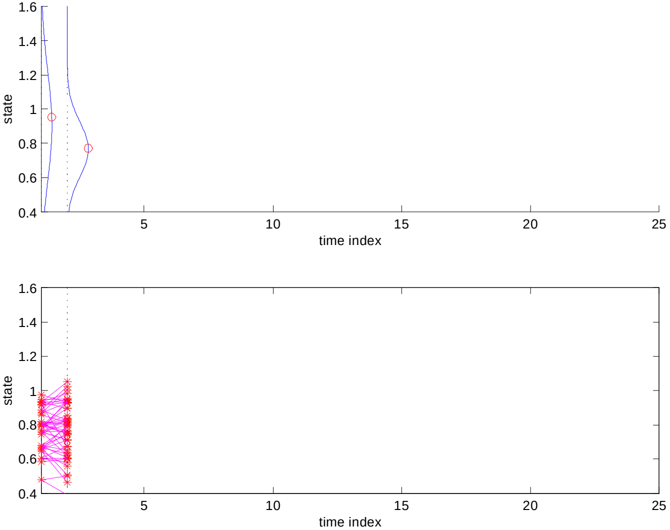


Figure: $p\left(x_1 \mid y_1\right)$, $p\left(x_2 \mid y_{1:2}\right)$and $\widehat{\mathbb{E}}\left[X_1 \mid y_1\right]$, $\widehat{\mathbb{E}}\left[X_2 \mid y_{1:2}\right]$ (top) and particle approximation of $p\left(x_{1:2} \mid y_{1:2}\right)$ (bottom)

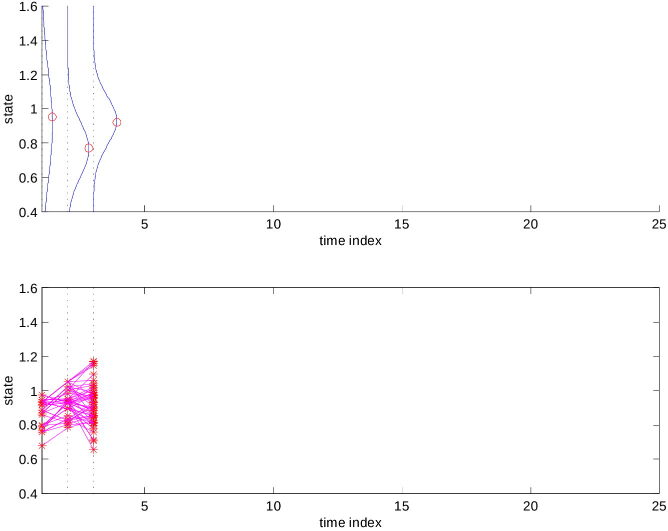


Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, 2, 3$ (top) and particle approximation of $p\left(x_{1:3} \middle| y_{1:3}\right)$ (bottom)

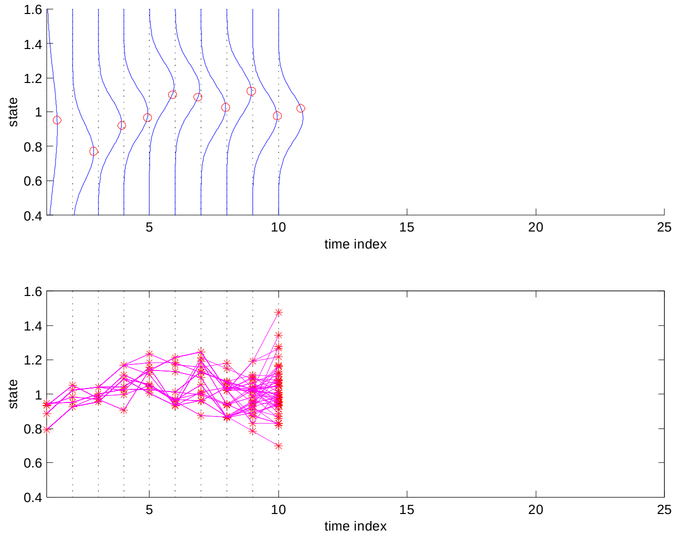


Figure: $p\left(x_k \mid y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \mid y_{1:k}\right]$ for $k = 1, ..., 10$ (top) and particle approximation of $p\left(x_{1:10} \mid y_{1:10}\right)$ (bottom)

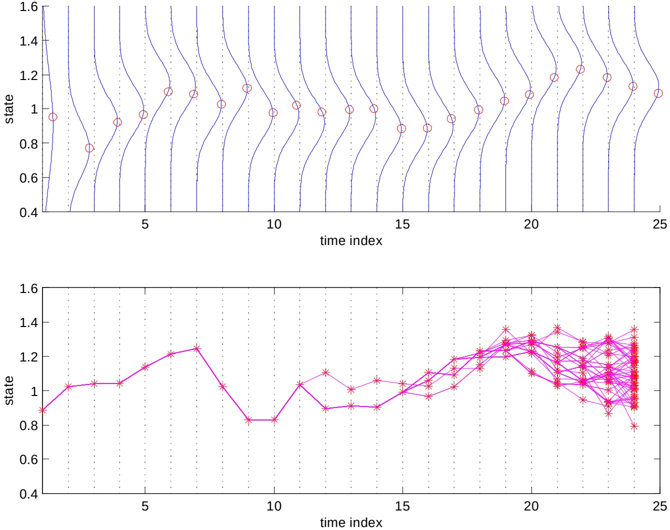


Figure: $p\left(x_k \mid y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \mid y_{1:k}\right]$ for $k = 1, ..., 24$ (top) and particle approximation of $p\left(x_{1:24} \mid y_{1:24}\right)$ (bottom)

Illustration of the Degeneracy Problem

- **Degeneracy problem**. For any N and any k, there exists $n(k, N)$ such that for any $n \geq n(k, N)$

$$\widehat{p}\left(x_{1:k} \middle| y_{1:n}\right) = \delta_{X^*_{1:k}}\left(x_{1:k}\right).$$

$\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right)$ is an unreliable approximation of $p\left(x_{1:n} \middle| y_{1:n}\right)$ as $n \nearrow$.

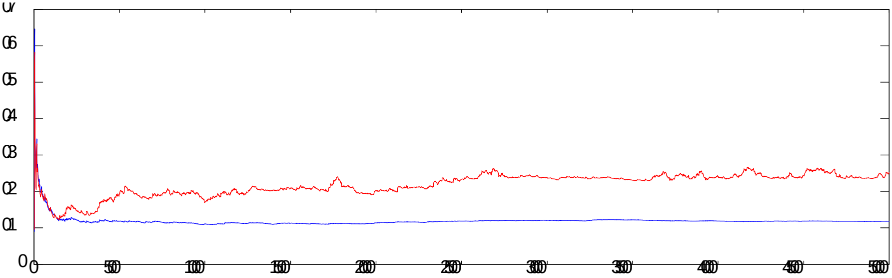


Figure: Exact calculation of $\frac{1}{n}\mathbb{E}\left[\sum_{k=1}^{n} X_k \middle| y_{1:n}\right]$ via Kalman (blue) vs SMC estimate (red) for $N = 1000$. As n increases, the SMC estimate deteriorates.

Convergence Results

- Numerous precise convergence results are available for SMC methods (Del Moral, 2004).

Convergence Results

- Numerous precise convergence results are available for SMC methods (Del Moral, 2004).
- Let $\varphi_n : \mathcal{X}^n \rightarrow \mathbb{R}$ and consider

$$\overline{\varphi}_n = \int \varphi_n \left(x_{1:n} \right) p \left(\left. x_{1:n} \right| y_{1:n} \right) dx_{1:n},$$

$$\widehat{\varphi}_n = \int \varphi_n \left(x_{1:n} \right) \widehat{p} \left(\left. x_{1:n} \right| y_{1:n} \right) dx_{1:n} = \frac{1}{N} \sum_{i=1}^{N} \varphi_n \left(X_{1:n}^{(i)} \right).$$

Convergence Results

- Numerous precise convergence results are available for SMC methods (Del Moral, 2004).
- Let $\varphi_n : \mathcal{X}^n \rightarrow \mathbb{R}$ and consider

$$\overline{\varphi}_n = \int \varphi_n(x_{1:n})\, p(x_{1:n}|\, y_{1:n})\, dx_{1:n},$$

$$\widehat{\varphi}_n = \int \varphi_n(x_{1:n})\, \widehat{p}(x_{1:n}|\, y_{1:n})\, dx_{1:n} = \frac{1}{N}\sum_{i=1}^{N} \varphi_n\left(X_{1:n}^{(i)}\right).$$

- Under very weak assumptions, we have for any $p > 0$

$$\mathbb{E}\left[\left|\widehat{\varphi}_n - \overline{\varphi}_n\right|^p\right]^{1/p} \leq \frac{C_n}{\sqrt{N}}$$

and

$$\lim_{N\rightarrow\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right).$$

Convergence Results

- Numerous precise convergence results are available for SMC methods (Del Moral, 2004).
- Let $\varphi_n : \mathcal{X}^n \rightarrow \mathbb{R}$ and consider

$$\overline{\varphi}_n = \int \varphi_n(x_{1:n})\, p(x_{1:n}|\, y_{1:n})\, dx_{1:n},$$

$$\widehat{\varphi}_n = \int \varphi_n(x_{1:n})\, \widehat{p}(x_{1:n}|\, y_{1:n})\, dx_{1:n} = \frac{1}{N}\sum_{i=1}^{N} \varphi_n\left(X_{1:n}^{(i)}\right).$$

- Under very weak assumptions, we have for any $p > 0$

$$\mathbb{E}\left[\left|\widehat{\varphi}_n - \overline{\varphi}_n\right|^p\right]^{1/p} \leq \frac{C_n}{\sqrt{N}}$$

and

$$\lim_{N\rightarrow\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right).$$

- **Very weak results**: C_n and σ_n^2 can increase with n and will for a path-dependent $\varphi_n(x_{1:n})$ as the degeneracy problem suggests!

Stronger Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1 \right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1' \right) \right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

Stronger Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n\right| y_{2:n}, X_1 = x_1) - p\left(x_n\right| y_{2:n}, X_1 = x_1'\left.\right)\right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

- **Marginal distribution**. For $\varphi_n(x_{1:n}) = \varphi(x_n)$,

$$\mathbb{E}\left[\left|\widehat{\varphi}_n - \overline{\varphi}_n\right|^p\right]^{1/p} \leq \frac{C}{\sqrt{N}},$$

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \sigma_n^2 \leq D,$$

where C and D typically exponential in $\dim(X_n)$.

Stronger Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1\right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1'\right)\right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

- **Marginal distribution**. For $\varphi_n(x_{1:n}) = \varphi(x_n)$,

$$\mathbb{E}\left[\left|\widehat{\varphi}_n - \overline{\varphi}_n\right|^p\right]^{1/p} \leq \frac{C}{\sqrt{N}},$$

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \sigma_n^2 \leq D,$$

  where C and D typically exponential in $\dim(X_n)$.

- **Marginal likelihood.**

$$\lim_{N\to\infty} \sqrt{N}\left(\log \widehat{p}\left(y_{1:n}\right) - \log p\left(y_{1:n}\right)\right) \Rightarrow \mathcal{N}\left(0, \overline{\sigma}_n^2\right) \text{ with } \overline{\sigma}_n^2 \leq A\ n.$$

Stronger Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1 \right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1' \right) \right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

- **Marginal distribution**. For $\varphi_n(x_{1:n}) = \varphi(x_n)$,

$$\mathbb{E}\left[\left| \widehat{\varphi}_n - \overline{\varphi}_n \right|^p \right]^{1/p} \leq \frac{C}{\sqrt{N}},$$

$$\lim_{N \to \infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n \right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \sigma_n^2 \leq D,$$

  where C and D typically exponential in $\dim(X_n)$.

- **Marginal likelihood.**

$$\lim_{N \to \infty} \sqrt{N}\left(\log \widehat{p}\left(y_{1:n}\right) - \log p\left(y_{1:n}\right) \right) \Rightarrow \mathcal{N}\left(0, \overline{\sigma}_n^2\right) \text{ with } \overline{\sigma}_n^2 \leq A\, n.$$

- **Resampling is necessary.** Without resampling, we have $\log \widehat{p}\left(y_{1:n}\right) = \log \frac{1}{N}\sum_{i=1}^N \prod_{k=1}^n g\left(y_k \middle| \widetilde{X}_k^{(i)} \right)$ which has a variance increasing exponentially with n even for trivial examples.

Improving the Sampling Step

- **Boostrap filter**. Very inefficient for vague prior/peaky likelihood; e.g. $p(x_{n-1}|\, y_{1:n-1}) = \mathcal{N}(x_{n-1}; m, \sigma^2)$, $f(x_n|\, x_{n-1}) = \mathcal{N}(x_n; x_{n-1}, \sigma_v^2)$ and $g(y_n|\, x_n) = \mathcal{N}(y_n; x_n, \sigma_w^2)$.

Improving the Sampling Step

- **Boostrap filter**. Very inefficient for vague prior/peaky likelihood; e.g. $p\left(x_{n-1}\middle|\, y_{1:n-1}\right) = \mathcal{N}\left(x_{n-1}; m, \sigma^2\right)$, $f\left(x_n\middle|\, x_{n-1}\right) = \mathcal{N}\left(x_n; x_{n-1}, \sigma_v^2\right)$ and $g\left(y_n\middle|\, x_n\right) = \mathcal{N}\left(y_n; x_n, \sigma_w^2\right)$.
- **Optimal proposal/Perfect adaptation**. Resample $W_n \propto p\left(y_n\middle|\, x_{n-1}\right)$, sample $p\left(x_n\middle|\, y_n, x_{n-1}\right) \propto g\left(y_n\middle|\, x_n\right) f\left(x_n\middle|\, x_{n-1}\right)$.

Improving the Sampling Step

- **Boostrap filter**. Very inefficient for vague prior/peaky likelihood; e.g. $p\left(x_{n-1} \middle| y_{1:n-1}\right) = \mathcal{N}\left(x_{n-1}; m, \sigma^2\right)$, $f\left(x_n \middle| x_{n-1}\right) = \mathcal{N}\left(x_n; x_{n-1}, \sigma_v^2\right)$ and $g\left(y_n \middle| x_n\right) = \mathcal{N}\left(y_n; x_n, \sigma_w^2\right)$.
- **Optimal proposal/Perfect adaptation**. Resample $W_n \propto p\left(y_n \middle| x_{n-1}\right)$, sample $p\left(x_n \middle| y_n, x_{n-1}\right) \propto g\left(y_n \middle| x_n\right) f\left(x_n \middle| x_{n-1}\right)$.

- **Boostrap filter**. Very inefficient for vague prior/peaky likelihood; e.g. $p(x_{n-1}|\, y_{1:n-1}) = \mathcal{N}(x_{n-1}; m, \sigma^2)$, $f(x_n|\, x_{n-1}) = \mathcal{N}(x_n; x_{n-1}, \sigma_v^2)$ and $g(y_n|\, x_n) = \mathcal{N}(y_n; x_n, \sigma_w^2)$.
- **Optimal proposal/Perfect adaptation**. Resample $W_n \propto p(y_n|\, x_{n-1})$, sample $p(x_n|\, y_n, x_{n-1}) \propto g(y_n|\, x_n)\, f(x_n|\, x_{n-1})$.

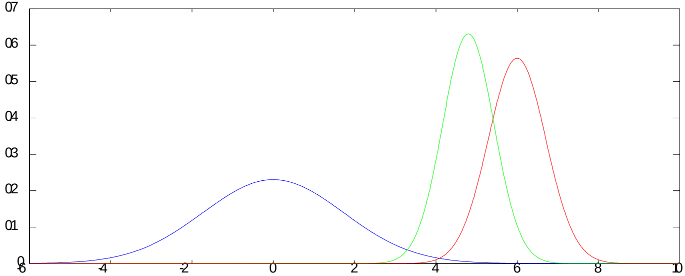


Figure: $p(x_n|\, y_{1:n-1}) = \int f(x_n|\, x_{n-1})\, p(x_{n-1}|\, y_{1:n-1})\, dx_{n-1}$ (blue), $\int p(x_n|\, y_n, x_{n-1})\, p(x_{n-1}|\, y_{1:n-1})\, dx_{n-1}$ (green), $g(y_n|\, x_n)$ (red)

Various standard improvements

- **Approximate optimal proposal**. Design analytical approximation via EKF, UKF $\widehat{p}(x_n | y_n, x_{n-1})$ of $p(x_n | y_n, x_{n-1})$. Sample $\widehat{p}(x_n | y_n, x_{n-1})$ and set

$$W_n \propto \frac{g(y_n | x_n) f(x_n | x_{n-1})}{\widehat{p}(x_n | y_n, x_{n-1})};$$

see also Auxiliary Particle Filters (Pitt & Shephard, 1999)

Various standard improvements

- **Approximate optimal proposal**. Design analytical approximation via EKF, UKF $\widehat{p}\left(x_n \mid y_n, x_{n-1}\right)$ of $p\left(x_n \mid y_n, x_{n-1}\right)$. Sample $\widehat{p}\left(x_n \mid y_n, x_{n-1}\right)$ and set

$$W_n \propto \frac{g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right)}{\widehat{p}\left(x_n \mid y_n, x_{n-1}\right)};$$

  see also Auxiliary Particle Filters (Pitt & Shephard, 1999)

- **Resample Move** (Gilks & Berzuini, 1999). After the resampling step, you have $X_{1:n}^{(i)} = X_{1:n}^{(j)}$ for $i \neq j$. To add diversity among particles, use an MCMC kernel $X_{1:n}^{\prime(i)} \sim K_n\left(x_{1:n} \middle| X_{1:n}^{(i)}\right)$ where

$$p\left(x'_{1:n} \mid y_{1:n}\right) = \int p\left(x_{1:n} \mid y_{1:n}\right) K_n\left(x'_{1:n} \mid x_{1:n}\right) dx_{1:n}$$

  Here K_n does not have to be ergodic.

Improving the Resampling Step

- Resample N times $X_{1:n}^{(i)} \sim \widetilde{p}\left(x_{1:n} \middle| y_{1:n} \right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n} \right)$ to obtain $\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right)$ is called *multinomial resampling* as

$$\widehat{p}\left(x_{1:n} \middle| y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n} \right) = \sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n} \right)$$

where $\left\{ N_n^{(i)} \right\}$ follow a multinomial with $\mathbb{E}\left[N_n^{(i)} \right] = N W_n^{(i)}$, $\mathbb{V}\left[N_n^{(1)} \right] = N W_n^{(i)} \left(1 - W_n^{(i)} \right)$.

Improving the Resampling Step

- Resample N times $X_{1:n}^{(i)} \sim \widetilde{p}\left(\left. x_{1:n} \right| y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right)$ to obtain $\widehat{p}\left(\left. x_{1:n} \right| y_{1:n}\right)$ is called *multinomial resampling* as

$$\widehat{p}\left(\left. x_{1:n} \right| y_{1:n}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right) = \sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{\widetilde{X}_{1:n}^{(i)}}\left(x_{1:n}\right)$$

  where $\left\{N_n^{(i)}\right\}$ follow a multinomial with $\mathbb{E}\left[N_n^{(i)}\right] = NW_n^{(i)}$, $\mathbb{V}\left[N_n^{(1)}\right] = NW_n^{(i)}\left(1 - W_n^{(i)}\right)$.

- Better resampling steps can be designed with $\mathbb{E}\left[N_n^{(i)}\right] = NW_n^{(i)}$ but smaller $\mathbb{V}\left[N_n^{(i)}\right]$; e.g. stratified resampling (Kitagawa, 1996).

Online Bayesian Parameter Estimation

- Assume we have

$$
\begin{aligned}
X_n \mid (X_{n-1} = x_{n-1}) &\sim f_\theta (x_n \mid x_{n-1}) , \\
Y_n \mid (X_n = x_n) &\sim g_\theta (y_n \mid x_n) ,
\end{aligned}
$$

where θ is an *unknown* static parameter with prior $p(\theta)$.

Online Bayesian Parameter Estimation

- Assume we have

$$X_n | (X_{n-1} = x_{n-1}) \sim f_\theta (x_n | x_{n-1}),$$
$$Y_n | (X_n = x_n) \sim g_\theta (y_n | x_n),$$

where θ is an *unknown* static parameter with prior $p(\theta)$.

- Given data $y_{1:n}$, inference relies on

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) p_\theta (x_{1:n} | y_{1:n})$$

where

$$p(\theta | y_{1:n}) \propto p_\theta (y_{1:n}) p(\theta).$$

Online Bayesian Parameter Estimation

- Assume we have

$$X_n | (X_{n-1} = x_{n-1}) \sim f_\theta (x_n | x_{n-1}),$$
$$Y_n | (X_n = x_n) \sim g_\theta (y_n | x_n),$$

  where θ is an *unknown* static parameter with prior $p(\theta)$.

- Given data $y_{1:n}$, inference relies on

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) \, p_\theta (x_{1:n} | y_{1:n})$$

  where

$$p(\theta | y_{1:n}) \propto p_\theta (y_{1:n}) \, p(\theta).$$

- SMC methods apply as it is a standard model with extended state $Z_n = (X_n, \theta_n)$ where

$$f(z_n | z_{n-1}) = \underbrace{\delta_{\theta_{n-1}} (\theta_n)}_{\text{practical problems}} \, f_{\theta_n} (x_n | x_{n-1}), \; g(y_n | z_n) = g_\theta (y_n | x_n).$$

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\log \widehat{p}_{\theta}\left(y_{1:n}\right)\right]$ is in Cn/N. In a Bayesian context, the problem is even more severe as $p\left(\theta \mid y_{1:n}\right) \propto p_{\theta}\left(y_{1:n}\right) p\left(\theta\right)$. Exponential stability assumption cannot hold as $\theta_n = \theta_1$.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\log \widehat{p}_{\theta}\left(y_{1:n}\right)\right]$ is in Cn/N. In a Bayesian context, the problem is even more severe as $p\left(\theta \middle| y_{1:n}\right) \propto p_{\theta}\left(y_{1:n}\right) p\left(\theta\right)$. Exponential stability assumption cannot hold as $\theta_n = \theta_1$.
- To mitigate but **NOT** solve the problem, introduce MCMC steps on θ; e.g. (Andrieu, D.&D.,1999; Fearnhead, 1998, 2002; Gilks & Berzuini 1999,2001,2003; Storvik, 2002; Polson & Johannes, 2007; Vercauteren et al., 2005).

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\log \widehat{p}_{\theta}\left(y_{1:n}\right)\right]$ is in Cn/N. In a Bayesian context, the problem is even more severe as $p\left(\theta \mid y_{1:n}\right) \propto p_{\theta}\left(y_{1:n}\right) p\left(\theta\right)$. Exponential stability assumption cannot hold as $\theta_n = \theta_1$.
- To mitigate but **NOT** solve the problem, introduce MCMC steps on θ; e.g. (Andrieu, D.&D.,1999; Fearnhead, 1998, 2002; Gilks & Berzuini 1999,2001,2003; Storvik, 2002; Polson & Johannes, 2007; Vercauteren et al., 2005).
- When $p\left(\theta \mid y_{1:n}, x_{1:n}\right) = p\left(\theta \mid s_n\left(x_{1:n}, y_{1:n}\right)\right)$ where $s_n\left(x_{1:n}, y_{1:n}\right)$ is fixed-dimensional, this is an elegant algorithm but still relies on $\widehat{p}\left(x_{1:n} \mid y_{1:n}\right)$ so degeneracy will creep in.

Cautionary Warning

- For fixed θ, $\mathbb{V}\left[\log \widehat{p}_{\theta}\left(y_{1:n}\right)\right]$ is in Cn/N. In a Bayesian context, the problem is even more severe as $p\left(\theta \mid y_{1:n}\right) \propto p_{\theta}\left(y_{1:n}\right) p\left(\theta\right)$. Exponential stability assumption cannot hold as $\theta_n = \theta_1$.
- To mitigate but **NOT** solve the problem, introduce MCMC steps on θ; e.g. (Andrieu, D.&D.,1999; Fearnhead, 1998, 2002; Gilks & Berzuini 1999,2001,2003; Storvik, 2002; Polson & Johannes, 2007; Vercauteren et al., 2005).
- When $p\left(\theta \mid y_{1:n}, x_{1:n}\right) = p\left(\theta \mid s_n\left(x_{1:n}, y_{1:n}\right)\right)$ where $s_n\left(x_{1:n}, y_{1:n}\right)$ is fixed-dimensional, this is an elegant algorithm but still relies on $\widehat{p}\left(x_{1:n} \mid y_{1:n}\right)$ so degeneracy will creep in.
- As $\dim\left(Z_n\right) = \dim\left(X_n\right) + \dim\left(\theta\right)$, such methods are not recommended for high-dimensional θ, especially with vague priors.

Example of SMC with MCMC for Parameter Estimation

- Given at time $n-1$, the approximation at time n

$$\widehat{p}\left(\theta, x_{1:n-1} \middle| y_{1:n-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right)}\left(\theta, x_{1:n-1}\right).$$

Example of SMC with MCMC for Parameter Estimation

- Given at time $n-1$, the approximation at time n

$$\widehat{p}\left(\theta, x_{1:n-1} \middle| y_{1:n-1}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right)}\left(\theta, x_{1:n-1}\right).$$

- Sample $\widetilde{X}_n^{(i)} \sim f_{\theta_{n-1}^{(i)}}\left(x_n \middle| X_{n-1}^{(i)}\right)$, set $\widetilde{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_{1:n} \middle| y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\left(\theta_{n-1}^{(i)}, \widetilde{X}_{1:n}^{(i)}\right)}\left(x_{1:n}\right), \quad W_n^{(i)} \propto g_{\theta_{n-1}^{(i)}}\left(y_n \middle| \widetilde{X}_n^{(i)}\right).$$

Example of SMC with MCMC for Parameter Estimation

- Given at time $n-1$, the approximation at time n

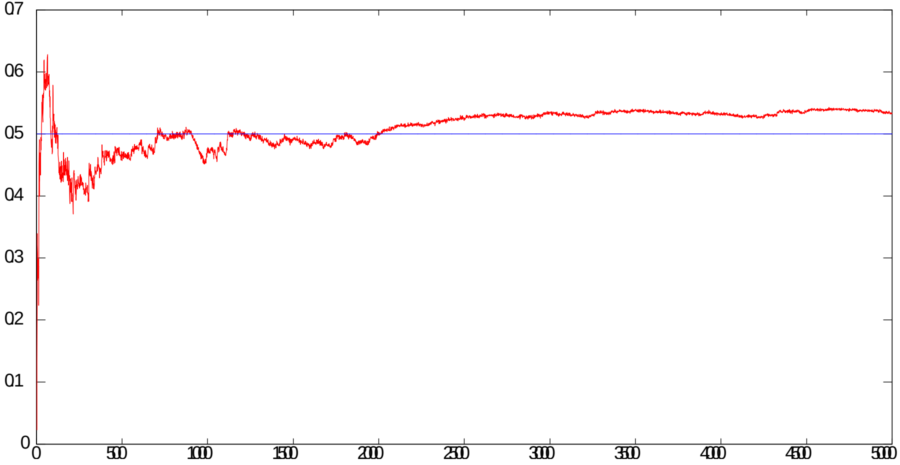
$$\widehat{p}\left(\theta, x_{1:n-1} \middle| y_{1:n-1}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\left(\theta_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right)}\left(\theta, x_{1:n-1}\right).$$

- Sample $\widetilde{X}_n^{(i)} \sim f_{\theta_{n-1}^{(i)}}\left(x_n \middle| X_{n-1}^{(i)}\right)$, set $\widetilde{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ and

$$\widetilde{p}\left(\theta, x_{1:n} \middle| y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\left(\theta_{n-1}^{(i)}, \widetilde{X}_{1:n}^{(i)}\right)}\left(x_{1:n}\right), \quad W_n^{(i)} \propto g_{\theta_{n-1}^{(i)}}\left(y_n \middle| \widetilde{X}_n^{(i)}\right).$$

- Resample $X_{1:n}^{(i)} \sim \widetilde{p}\left(x_{1:n} \middle| y_{1:n}\right)$ then sample $\theta_n^{(i)} \sim p\left(\theta \middle| y_{1:n}, X_{1:n}^{(i)}\right)$ to obtain $\widehat{p}\left(\theta, x_{1:n} \middle| y_{1:n}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\left(\theta_n^{(i)}, X_{1:n}^{(i)}\right)}\left(\theta, x_{1:n}\right)$.

Illustration of the Degeneracy Problem



SMC estimate of $\mathbb{E}\left[\theta \middle| y_{1:n}\right]$, as n increases the degeneracy creeps in.

Offline Bayesian Parameter Estimation

- Given data $y_{1:n}$, inference relies on

$$p\left(\theta, x_{1:n} \middle| y_{1:n}\right) = p\left(\theta \middle| y_{1:n}\right) p_{\theta}\left(x_{1:n} \middle| y_{1:n}\right)$$

  where

$$p\left(\theta \middle| y_{1:n}\right) \propto p_{\theta}\left(y_{1:n}\right) p\left(\theta\right).$$

Offline Bayesian Parameter Estimation

- Given data $y_{1:n}$, inference relies on

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) \, p_\theta(x_{1:n} | y_{1:n})$$

  where

$$p(\theta | y_{1:n}) \propto p_\theta(y_{1:n}) \, p(\theta).$$

- For a given θ, SMC can estimate both $p_\theta(x_{1:n} | y_{1:n})$ and $p_\theta(y_{1:n})$.

Offline Bayesian Parameter Estimation

- Given data $y_{1:n}$, inference relies on

$$p\left(\theta, x_{1:n} \middle| y_{1:n}\right) = p\left(\theta \middle| y_{1:n}\right) p_\theta\left(x_{1:n} \middle| y_{1:n}\right)$$

  where

$$p\left(\theta \middle| y_{1:n}\right) \propto p_\theta\left(y_{1:n}\right) p\left(\theta\right).$$

- For a given θ, SMC can estimate both $p_\theta\left(x_{1:n} \middle| y_{1:n}\right)$ and $p_\theta\left(y_{1:n}\right)$.
- Is it possible to use SMC within MCMC to sample from $p\left(\theta, x_{1:n} \middle| y_{1:n}\right)$?

Metropolis-Hastings (MH) Sampler

- To sample from a target $\pi(z)$, the MH sampler generates a Markov chain $\left\{Z^{(i)}\right\}$ according to the following mechanism. Given $Z^{(i-1)}$, propose a candidate $Z^{*} \sim q\left(z^{*} \middle| Z^{(i-1)}\right)$ and with probability

$$\alpha\left(Z^{(i-1)}, Z^{*}\right) = \min\left(1, \frac{\pi(Z^{*})\, q\left(Z^{(i-1)} \middle| Z^{*}\right)}{\pi\left(Z^{(i-1)}\right) q\left(Z^{*} \middle| Z^{(i-1)}\right)}\right)$$

set $Z^{(i)} = Z^{*}$, otherwise $Z^{(i)} = Z^{(i-1)}$.

Metropolis-Hastings (MH) Sampler

- To sample from a target $\pi(z)$, the MH sampler generates a Markov chain $\left\{Z^{(i)}\right\}$ according to the following mechanism. Given $Z^{(i-1)}$, propose a candidate $Z^* \sim q\left(z^* \middle| Z^{(i-1)}\right)$ and with probability

$$\alpha\left(Z^{(i-1)}, Z^*\right) = \min\left(1, \frac{\pi(Z^*)\, q\left(Z^{(i-1)} \middle| Z^*\right)}{\pi\left(Z^{(i-1)}\right) q\left(Z^* \middle| Z^{(i-1)}\right)}\right)$$

  set $Z^{(i)} = Z^*$, otherwise $Z^{(i)} = Z^{(i-1)}$.

- It can be easily shown that

$$\pi(z') = \int \pi(z)\, K\left(z' \middle| z\right) dz$$

  where $K\left(z' \middle| z\right)$ is the transition kernel of the MH and under weak assumptions $Z^{(i)} \sim \pi(z)$ as $i \to \infty$.

Marginal Metropolis-Hastings Sampler

- Consider the following so-called marginal MH algorithm which target

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) \, p_\theta(x_{1:n} | y_{1:n})$$

using the proposal

$$q((x^*_{1:n}, \theta^*) | (x_{1:n}, \theta)) = q(\theta^* | \theta) \, p_{\theta^*}(x^*_{1:n} | y_{1:n}) .$$

Marginal Metropolis-Hastings Sampler

- Consider the following so-called marginal MH algorithm which target

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) \, p_\theta(x_{1:n} | y_{1:n})$$

using the proposal

$$q((x_{1:n}^*, \theta^*) | (x_{1:n}, \theta)) = q(\theta^* | \theta) \, p_{\theta^*}(x_{1:n}^* | y_{1:n}).$$

- The MH acceptance probability is

$$\begin{aligned} &\min\left(1, \frac{p(\theta^*, x_{1:n}^* | y_{1:n})}{p(\theta, x_{1:n} | y_{1:n})} \frac{q((x_{1:n}, \theta) | (x_{1:n}^*, \theta^*))}{q((x_{1:n}^*, \theta^*) | (x_{1:n}, \theta))}\right) \\ &= \min\left(1, \frac{p_{\theta^*}(y_{1:n}) \, p(\theta^*)}{p_\theta(y_{1:n}) \, p(\theta)} \frac{q(\theta | \theta^*)}{q(\theta^* | \theta)}\right) \end{aligned}$$

Marginal Metropolis-Hastings Sampler

- Consider the following so-called marginal MH algorithm which target

$$p(\theta, x_{1:n} | y_{1:n}) = p(\theta | y_{1:n}) \, p_\theta(x_{1:n} | y_{1:n})$$

using the proposal

$$q((x^*_{1:n}, \theta^*) | (x_{1:n}, \theta)) = q(\theta^* | \theta) \, p_{\theta^*}(x^*_{1:n} | y_{1:n}).$$

- The MH acceptance probability is

$$\begin{aligned}
&\min\left(1, \frac{p(\theta^*, x^*_{1:n} | y_{1:n})}{p(\theta, x_{1:n} | y_{1:n})} \frac{q((x_{1:n}, \theta) | (x^*_{1:n}, \theta^*))}{q((x^*_{1:n}, \theta^*) | (x_{1:n}, \theta))}\right) \\
&= \min\left(1, \frac{p_{\theta^*}(y_{1:n}) \, p(\theta^*)}{p_\theta(y_{1:n}) \, p(\theta)} \frac{q(\theta | \theta^*)}{q(\theta^* | \theta)}\right)
\end{aligned}$$

- **Problem**: We do not know $p_\theta(y_{1:n})$ analytically and cannot sample from $p_\theta(x_{1:n} | y_{1:n})$ so this algorithm cannot be implemented.

Marginal Metropolis-Hastings Sampler

- Consider the following so-called marginal MH algorithm which target

$$p\left(\theta, x_{1:n} \middle| y_{1:n}\right) = p\left(\theta \middle| y_{1:n}\right) p_{\theta}\left(x_{1:n} \middle| y_{1:n}\right)$$

  using the proposal

$$q\left(\left(x_{1:n}^{*}, \theta^{*}\right) \middle| \left(x_{1:n}, \theta\right)\right) = q\left(\theta^{*} \middle| \theta\right) p_{\theta^{*}}\left(x_{1:n}^{*} \middle| y_{1:n}\right).$$

- The MH acceptance probability is

$$\begin{aligned} &\min\left(1, \frac{p\left(\theta^{*}, x_{1:n}^{*} \middle| y_{1:n}\right)}{p\left(\theta, x_{1:n} \middle| y_{1:n}\right)} \frac{q\left(\left(x_{1:n}, \theta\right) \middle| \left(x_{1:n}^{*}, \theta^{*}\right)\right)}{q\left(\left(x_{1:n}^{*}, \theta^{*}\right) \middle| \left(x_{1:n}, \theta\right)\right)}\right) \\ &= \min\left(1, \frac{p_{\theta^{*}}\left(y_{1:n}\right) p\left(\theta^{*}\right)}{p_{\theta}\left(y_{1:n}\right) p\left(\theta\right)} \frac{q\left(\theta \middle| \theta^{*}\right)}{q\left(\theta^{*} \middle| \theta\right)}\right) \end{aligned}$$

- **Problem**: We do not know $p_{\theta}\left(y_{1:n}\right)$ analytically and cannot sample from $p_{\theta}\left(x_{1:n} \middle| y_{1:n}\right)$ so this algorithm cannot be implemented.
- **"Idea"**: Use SMC approximations of $p_{\theta}\left(x_{1:n} \middle| y_{1:n}\right)$ and $p_{\theta}\left(y_{1:n}\right)$.

Particle Marginal MH Sampler

- At iteration i, given $\{\theta(i-1), X_{1:n}(i-1), \widehat{p}_{\theta(i-1)}(y_{1:n})\}$ then sample $\theta^* \sim q(\theta|\theta(i-1))$, run an SMC algorithm to obtain $\widehat{p}_{\theta^*}(x_{1:n}|y_{1:n})$ and $\widehat{p}_{\theta^*}(y_{1:n})$.

Particle Marginal MH Sampler

- At iteration i, given $\left\{\theta(i-1), X_{1:n}(i-1), \widehat{p}_{\theta(i-1)}(y_{1:n})\right\}$ then sample $\theta^{*} \sim q(\theta|\, \theta(i-1))$, run an SMC algorithm to obtain $\widehat{p}_{\theta^{*}}(x_{1:n}|\, y_{1:n})$ and $\widehat{p}_{\theta^{*}}(y_{1:n})$.
- Sample $X_{1:n}^{*} \sim \widehat{p}_{\theta^{*}}(x_{1:n}|\, y_{1:n})$.

Particle Marginal MH Sampler

- At iteration i, given $\left\{\theta(i-1), X_{1:n}(i-1), \widehat{p}_{\theta(i-1)}(y_{1:n})\right\}$ then sample $\theta^* \sim q(\theta|\,\theta(i-1))$, run an SMC algorithm to obtain $\widehat{p}_{\theta^*}(x_{1:n}|\,y_{1:n})$ and $\widehat{p}_{\theta^*}(y_{1:n})$.
- Sample $X_{1:n}^* \sim \widehat{p}_{\theta^*}(x_{1:n}|\,y_{1:n})$.
- With probability

$$\min\left(1, \frac{\widehat{p}_{\theta^*}(y_{1:n})\,p(\theta^*)}{\widehat{p}_{\theta(i-1)}(y_{1:n})\,p(\theta(i-1))}\frac{q(\theta(i-1)|\,\theta^*)}{q(\theta^*|\,\theta(i-1))}\right)$$

set $\left\{\theta(i), X_{1:n}(i), \widehat{p}_{\theta(i)}(y_{1:n})\right\} = \left\{\theta^*, X_{1:n}^*, \widehat{p}_{\theta^*}(y_{1:n})\right\}$ otherwise set $\left\{\theta(i), X_{1:n}(i), \widehat{p}_{\theta(i)}(y_{1:n})\right\} = \left\{\theta(i-1), X_{1:n}(i-1), \widehat{p}_{\theta(i-1)}(y_{1:n})\right\}$.

Validity of the Particle Marginal MH Sampler

- This algorithm (without sampling $X_{1:n}$) was proposed as an approximate MCMC algorithm to sample from $p(\theta | y_{1:n})$ in (Fernandez-Villaverde & Rubio-Ramirez, 2007).

Validity of the Particle Marginal MH Sampler

- This algorithm (without sampling $X_{1:n}$) was proposed as an approximate MCMC algorithm to sample from $p(\theta | y_{1:n})$ in (Fernandez-Villaverde & Rubio-Ramirez, 2007).
- Whatever being $N \geq 1$, this algorithm admits exactly $p(\theta, x_{1:n} | y_{1:n})$ as invariant distribution (Andrieu, D. & Holenstein, 2010). A particle version of the Gibbs sampler also exists.

Validity of the Particle Marginal MH Sampler

- This algorithm (without sampling $X_{1:n}$) was proposed as an approximate MCMC algorithm to sample from $p(\theta | y_{1:n})$ in (Fernandez-Villaverde & Rubio-Ramirez, 2007).
- Whatever being $N \geq 1$, this algorithm admits exactly $p(\theta, x_{1:n} | y_{1:n})$ as invariant distribution (Andrieu, D. & Holenstein, 2010). A particle version of the Gibbs sampler also exists.
- The higher N, the better the performance of the algorithm: N scales roughly linearly with n.

Validity of the Particle Marginal MH Sampler

- This algorithm (without sampling $X_{1:n}$) was proposed as an approximate MCMC algorithm to sample from $p(\theta | y_{1:n})$ in (Fernandez-Villaverde & Rubio-Ramirez, 2007).
- Whatever being $N \geq 1$, this algorithm admits exactly $p(\theta, x_{1:n} | y_{1:n})$ as invariant distribution (Andrieu, D. & Holenstein, 2010). A particle version of the Gibbs sampler also exists.
- The higher N, the better the performance of the algorithm: N scales roughly linearly with n.
- Particularly useful in scenarios where X_n moderate dimensional & θ high dimensional. Admits the plug and play property (Ionides et al., 2006).

Inference for Stochastic Kinetic Models

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\begin{aligned}
\Pr\left(X_{t+dt}^1 = x_t^1 + 1, X_{t+dt}^2 = x_t^2 \,\middle|\, x_t^1, x_t^2\right) &= \alpha\, x_t^1\, dt + o\left(dt\right), \\
\Pr\left(X_{t+dt}^1 = x_t^1 - 1, X_{t+dt}^2 = x_t^2 + 1 \,\middle|\, x_t^1, x_t^2\right) &= \beta\, x_t^1\, x_t^2\, dt + o\left(dt\right), \\
\Pr\left(X_{t+dt}^1 = x_t^1, X_{t+dt}^2 = x_t^2 - 1 \,\middle|\, x_t^1, x_t^2\right) &= \gamma\, x_t^2\, dt + o\left(dt\right),
\end{aligned}$$

observed at discrete times

$$Y_n = X_{n\Delta}^1 + W_n \text{ with } W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

Inference for Stochastic Kinetic Models

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\begin{aligned}
&\Pr\left(X_{t+dt}^1 = x_t^1 + 1, X_{t+dt}^2 = x_t^2 \,\middle|\, x_t^1, x_t^2\right) = \alpha\, x_t^1\, dt + o\left(dt\right),\\
&\Pr\left(X_{t+dt}^1 = x_t^1 - 1, X_{t+dt}^2 = x_t^2 + 1 \,\middle|\, x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2\, dt + o\left(dt\right),\\
&\Pr\left(X_{t+dt}^1 = x_t^1, X_{t+dt}^2 = x_t^2 - 1 \,\middle|\, x_t^1, x_t^2\right) = \gamma\, x_t^2\, dt + o\left(dt\right),
\end{aligned}$$

  observed at discrete times

$$Y_n = X_{n\Delta}^1 + W_n \text{ with } W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- We are interested in the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ a priori distributed as (Boys et al., 2008)

$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5).$$

Inference for Stochastic Kinetic Models

- Two species X_t^1 (prey) and X_t^2 (predator)

$$\Pr\left(X_{t+dt}^1 = x_t^1 + 1, X_{t+dt}^2 = x_t^2 \,\middle|\, x_t^1, x_t^2\right) = \alpha\, x_t^1\, dt + o\left(dt\right),$$
$$\Pr\left(X_{t+dt}^1 = x_t^1 - 1, X_{t+dt}^2 = x_t^2 + 1 \,\middle|\, x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2\, dt + o\left(dt\right),$$
$$\Pr\left(X_{t+dt}^1 = x_t^1, X_{t+dt}^2 = x_t^2 - 1 \,\middle|\, x_t^1, x_t^2\right) = \gamma\, x_t^2\, dt + o\left(dt\right),$$

observed at discrete times

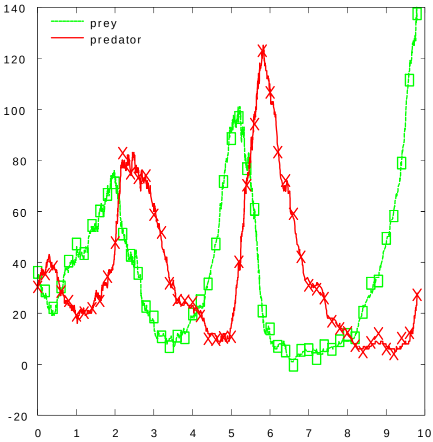
$$Y_n = X_{n\Delta}^1 + W_n \text{ with } W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right).$$

- We are interested in the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ a priori distributed as (Boys et al., 2008)

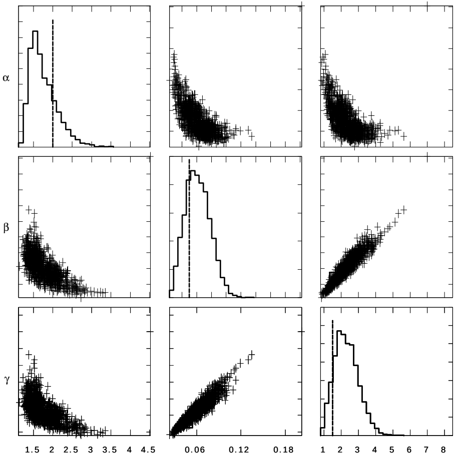
$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5).$$

- MCMC methods require reversible jumps, PMMH requires only forward simulation.

Experimental Results



Simulated data



Posterior distributions

SMC Fixed-Lag Smoothing Approximation

- Direct SMC approximations of $p(x_{1:n}|y_{1:n})$ and its marginals $p(x_k|y_{1:n})$ gets poorer as $n \nearrow$.

SMC Fixed-Lag Smoothing Approximation

- Direct SMC approximations of $p\left(x_{1:n} \mid y_{1:n}\right)$ and its marginals $p\left(x_k \mid y_{1:n}\right)$ gets poorer as $n \nearrow$.
- The fixed-lag smoothing approximation (Kitagawa & Sato, 2001) relies on

$$p\left(x_{1:k} \mid y_{1:n}\right) \approx p\left(x_{1:k} \mid y_{1:k+\Delta}\right) \text{ for } \Delta \text{ large enough.}$$

SMC Fixed-Lag Smoothing Approximation

- Direct SMC approximations of $p(x_{1:n}|\, y_{1:n})$ and its marginals $p(x_k|\, y_{1:n})$ gets poorer as $n \nearrow$.
- The fixed-lag smoothing approximation (Kitagawa & Sato, 2001) relies on

$$p(x_{1:k}|\, y_{1:n}) \approx p(x_{1:k}|\, y_{1:k+\Delta}) \text{ for } \Delta \text{ large enough.}$$

- **Algorithmically**: stop resampling $\left\{X_k^{(i)}\right\}$ beyond time $k+\Delta$.

SMC Fixed-Lag Smoothing Approximation

- Direct SMC approximations of $p(x_{1:n} | y_{1:n})$ and its marginals $p(x_k | y_{1:n})$ gets poorer as $n \nearrow$.
- The fixed-lag smoothing approximation (Kitagawa & Sato, 2001) relies on

$$p(x_{1:k} | y_{1:n}) \approx p(x_{1:k} | y_{1:k+\Delta}) \text{ for } \Delta \text{ large enough.}$$

- **Algorithmically**: stop resampling $\left\{ X_k^{(i)} \right\}$ beyond time $k + \Delta$.
- Computational cost is $\mathcal{O}(Nn)$ but non-vanishing bias as $N \to \infty$ (Olsson & al., 2006).

SMC Fixed-Lag Smoothing Approximation

- Direct SMC approximations of $p(x_{1:n}|\, y_{1:n})$ and its marginals $p(x_k|\, y_{1:n})$ gets poorer as $n \nearrow$.
- The fixed-lag smoothing approximation (Kitagawa & Sato, 2001) relies on

$$p(x_{1:k}|\, y_{1:n}) \approx p(x_{1:k}|\, y_{1:k+\Delta}) \text{ for } \Delta \text{ large enough.}$$

- **Algorithmically**: stop resampling $\left\{X_k^{(i)}\right\}$ beyond time $k+\Delta$.
- Computational cost is $\mathcal{O}(Nn)$ but non-vanishing bias as $N \to \infty$ (Olsson & al., 2006).
- Picking Δ is difficult. Δ too small results in $p(x_{1:k}|\, y_{1:k+\Delta})$ being a poor approximation of $p(x_{1:k}|\, y_{1:n})$. Δ too large improves the approximation but particle degeneracy creeps in.

SMC Forward Filtering Backward Smoothing

- **Forward filtering Backward smoothing** (FFBS).

$$\overbrace{p\left(x_k \mid y_{1:n}\right)}^{\text{smoother at } k} = \int \overbrace{p\left(x_{k+1} \mid y_{1:n}\right)}^{\text{smoother at } k+1} \underbrace{\frac{f\left(x_{k+1} \mid x_k\right) \overbrace{p\left(x_k \mid y_{1:k}\right)}^{\text{filter at } k}}{p\left(x_{k+1} \mid y_{1:n}\right)}}_{\text{backward transition } p\left(x_k \mid y_{1:n}, x_{k+1}\right)} dx_{k+1}$$

SMC Forward Filtering Backward Smoothing

- **Forward filtering Backward smoothing** (FFBS).

$$\overbrace{p\left(x_k \middle| y_{1:n}\right)}^{\text{smoother at } k} = \int \overbrace{p\left(x_{k+1} \middle| y_{1:n}\right)}^{\text{smoother at } k+1} \underbrace{\frac{f\left(x_{k+1} \middle| x_k\right) \overbrace{p\left(x_k \middle| y_{1:k}\right)}^{\text{filter at } k}}{p\left(x_{k+1} \middle| y_{1:n}\right)}}_{\text{backward transition } p(x_k | y_{1:n}, x_{k+1})} dx_{k+1}$$

- **SMC Implementation**: For $k = 1, ..., n$, compute $\widehat{p}\left(x_k \middle| y_{1:k}\right)$. For $k = n-1, ..., 1$, compute $\widehat{p}\left(x_k \middle| y_{1:n}\right) = \sum_{i=1}^{N} W_{k|n}^{(i)} \delta_{X_k^{(i)}}\left(x_k\right)$ with cost $\mathcal{O}\left(N^2 n\right)$ using

$$W_{k|n}^{(i)} = \sum_{j=1}^{N} W_{k+1|n}^{(j)} \frac{f\left(X_{k+1}^{(j)} | X_k^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{k+1}^{(j)} | X_k^{(l)}\right)}.$$

SMC Forward Filtering Backward Smoothing

- **Forward filtering Backward smoothing** (FFBS).

$$\overbrace{p\left(x_k \mid y_{1:n}\right)}^{\text{smoother at } k} = \int \overbrace{p\left(x_{k+1} \mid y_{1:n}\right)}^{\text{smoother at } k+1} \underbrace{\frac{f\left(x_{k+1} \mid x_k\right) \overbrace{p\left(x_k \mid y_{1:k}\right)}^{\text{filter at } k}}{p\left(x_{k+1} \mid y_{1:n}\right)}}_{\text{backward transition } p\left(x_k \mid y_{1:n}, x_{k+1}\right)} dx_{k+1}$$

- **SMC Implementation**: For $k = 1, \ldots, n$, compute $\widehat{p}\left(x_k \mid y_{1:k}\right)$. For $k = n-1, \ldots, 1$, compute $\widehat{p}\left(x_k \mid y_{1:n}\right) = \sum_{i=1}^{N} W_{k|n}^{(i)} \delta_{X_k^{(i)}}\left(x_k\right)$ with cost $\mathcal{O}\left(N^2 n\right)$ using

$$W_{k|n}^{(i)} = \sum_{j=1}^{N} W_{k+1|n}^{(j)} \frac{f\left(X_{k+1}^{(j)} | X_k^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{k+1}^{(j)} | X_k^{(l)}\right)}.$$

- For $\varphi_n\left(x_{1:n}\right) = \sum_{k=1}^{n-1} s_k\left(x_k, x_{k+1}\right)$, the SMC FFBS estimates $\{\widehat{\varphi}_n\}$ can be computed online exactly (Del Moral, D. & Singh, 2009).

SMC Forward Filtering Backward Smoothing

- **Forward filtering Backward smoothing** (FFBS).

$$\overbrace{p\left(x_k \middle| y_{1:n}\right)}^{\text{smoother at } k} = \int \overbrace{p\left(x_{k+1} \middle| y_{1:n}\right)}^{\text{smoother at } k+1} \underbrace{\frac{f\left(x_{k+1} \middle| x_k\right) \overbrace{p\left(x_k \middle| y_{1:k}\right)}^{\text{filter at } k}}{p\left(x_{k+1} \middle| y_{1:n}\right)}}_{\text{backward transition } p\left(x_k \middle| y_{1:n}, x_{k+1}\right)} dx_{k+1}$$

- **SMC Implementation**: For $k = 1, \ldots, n$, compute $\widehat{p}\left(x_k \middle| y_{1:k}\right)$. For $k = n-1, \ldots, 1$, compute $\widehat{p}\left(x_k \middle| y_{1:n}\right) = \sum_{i=1}^{N} W_{k|n}^{(i)} \delta_{X_k^{(i)}}\left(x_k\right)$ with cost $\mathcal{O}\left(N^2 n\right)$ using

$$W_{k|n}^{(i)} = \sum_{j=1}^{N} W_{k+1|n}^{(j)} \frac{f\left(X_{k+1}^{(j)} | X_k^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{k+1}^{(j)} | X_k^{(l)}\right)}.$$

- For $\varphi_n\left(x_{1:n}\right) = \sum_{k=1}^{n-1} s_k\left(x_k, x_{k+1}\right)$, the SMC FFBS estimates $\left\{\widehat{\varphi}_n\right\}$ can be computed online exactly (Del Moral, D. & Singh, 2009).
- Sampling from $\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right)$ costs $O(Nn)$ (Godsill, D. & West, 2004) but $O(n)$ through rejection sampling (Douc et al., 2009).

SMC Generalized Two-Filter Smoothing

- **Generalized Two-Filter smoothing** (TFS)

$$p\left(x_k, x_{k+1} \middle| y_{1:n}\right) \propto \frac{\overbrace{p\left(x_k \middle| y_{1:k}\right)}^{\text{forward filter}} f\left(x_{k+1} \middle| x_k\right) \quad \overbrace{\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right)}^{\text{generalized backward filter}}}{\underbrace{\overline{p}\left(x_{k+1}\right)}_{\text{artificial prior}}},$$

$$\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right) \propto p\left(y_{k+1:n} \middle| x_{k+1}\right) \overline{p}\left(x_{k+1}\right).$$

SMC Generalized Two-Filter Smoothing

- **Generalized Two-Filter smoothing** (TFS)

$$p\left(x_k, x_{k+1} \middle| y_{1:n}\right) \propto \frac{\overbrace{p\left(x_k \middle| y_{1:k}\right)}^{\text{forward filter}} f\left(x_{k+1} \middle| x_k\right) \quad \overbrace{\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right)}^{\text{generalized backward filter}}}{\underbrace{\overline{p}\left(x_{k+1}\right)}_{\text{artificial prior}}},$$

$$\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right) \propto p\left(y_{k+1:n} \middle| x_{k+1}\right) \overline{p}\left(x_{k+1}\right).$$

- **SMC Implementation**: For $k = 1, ..., n$, compute $\widehat{p}\left(x_k \middle| y_{1:k}\right)$. For $k = n, ..., 1$, compute $\widehat{\overline{p}}\left(x_{k+1} \middle| y_{k+1:n}\right)$. Combine the forward and backward filters to obtain

$$\widehat{p}\left(x_k, x_{k+1} \middle| y_{1:n}\right) \propto \widehat{p}\left(x_k \middle| y_{1:k}\right) \frac{f\left(x_{k+1} \middle| x_k\right)}{\overline{p}\left(x_{k+1}\right)} \widehat{\overline{p}}\left(x_{k+1} \middle| y_{k+1:n}\right)$$

SMC Generalized Two-Filter Smoothing

- **Generalized Two-Filter smoothing** (TFS)

$$p\left(x_k, x_{k+1} \middle| y_{1:n}\right) \propto \frac{\overbrace{p\left(x_k \middle| y_{1:k}\right)}^{\text{forward filter}} f\left(x_{k+1} \middle| x_k\right) \overbrace{\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right)}^{\text{generalized backward filter}}}{\underbrace{\overline{p}\left(x_{k+1}\right)}_{\text{artificial prior}}},$$

$$\overline{p}\left(x_{k+1} \middle| y_{k+1:n}\right) \propto p\left(y_{k+1:n} \middle| x_{k+1}\right) \overline{p}\left(x_{k+1}\right).$$

- **SMC Implementation**: For $k = 1, \ldots, n$, compute $\widehat{p}\left(x_k \middle| y_{1:k}\right)$. For $k = n, \ldots, 1$, compute $\widehat{\overline{p}}\left(x_{k+1} \middle| y_{k+1:n}\right)$. Combine the forward and backward filters to obtain

$$\widehat{p}\left(x_k, x_{k+1} \middle| y_{1:n}\right) \propto \widehat{p}\left(x_k \middle| y_{1:k}\right) \frac{f\left(x_{k+1} \middle| x_k\right)}{\overline{p}\left(x_{k+1}\right)} \widehat{\overline{p}}\left(x_{k+1} \middle| y_{k+1:n}\right)$$

- Cost $\mathcal{O}\left(N^2 n\right)$ but $\mathcal{O}\left(Nn\right)$ through rejection sampling (Briers, D. & Maskell, 2008) and importance sampling (Fearnhead, Wyncoll & Tawn, 2008; Briers, D. & Singh, 2005).

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1\right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1'\right)\right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1 \right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1' \right) \right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

- **Additive functionals**. If $\varphi_n(x_{1:n}) = \sum_{k=1}^{n} \varphi(x_k)$, we have for the standard path-based SMC estimate (Poyiadjis, D. & Singh, 2009)

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \underline{A}n^2 \leq \sigma_n^2 \leq \overline{A}n^2.$$

For the FFBS and TFS estimates (Douc et al., 2009; Del Moral, D. & Singh, 2009), we have

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \sigma_n^2 \leq Cn$$

Convergence Results

- **Exponentially stability assumption**. For any x_1, x_1'

$$\frac{1}{2}\int \left| p\left(x_n \middle| y_{2:n}, X_1 = x_1\right) - p\left(x_n \middle| y_{2:n}, X_1 = x_1'\right)\right| dx_n \leq \alpha^n \text{ for } |\alpha| < 1.$$

- **Additive functionals**. If $\varphi_n(x_{1:n}) = \sum_{k=1}^n \varphi(x_k)$, we have for the standard path-based SMC estimate (Poyiadjis, D. & Singh, 2009)

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \underline{A}n^2 \leq \sigma_n^2 \leq \overline{A}n^2.$$

  For the FFBS and TFS estimates (Douc et al., 2009; Del Moral, D. & Singh, 2009), we have

$$\lim_{N\to\infty} \sqrt{N}\left(\widehat{\varphi}_n - \overline{\varphi}_n\right) \Rightarrow \mathcal{N}\left(0, \sigma_n^2\right) \text{ where } \sigma_n^2 \leq Cn$$

- Tradeoff between computational and statistical efficiency.

Experimental Results

- Consider a linear Gaussian model

$$X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \text{ and } X_k = \phi X_{k-1} + \sigma_V V_k, \;\; V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

$$Y_k = cX_k + \sigma_W W_k, \;\; W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

Experimental Results

- Consider a linear Gaussian model

$$X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \text{ and } X_k = \phi X_{k-1} + \sigma_V V_k, \;\; V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

$$Y_k = cX_k + \sigma_W W_k, \;\; W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We simulate 10,000 observations for $\theta = (\phi, \sigma_V, c, \sigma_W) = (0.8, 0.5, 1.0, 1.0)$.

Experimental Results

- Consider a linear Gaussian model

$$X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \text{ and } X_k = \phi X_{k-1} + \sigma_V V_k, \;\; V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

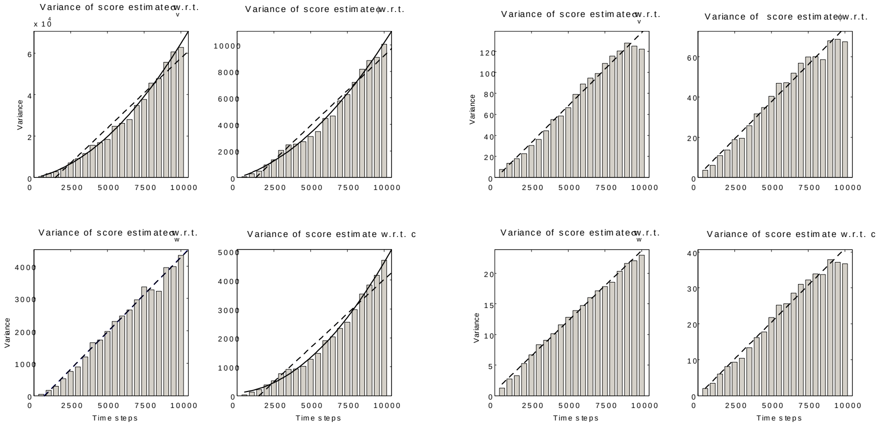
$$Y_k = cX_k + \sigma_W W_k, \;\; W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

- We simulate 10,000 observations for $\theta = (\phi, \sigma_V, c, \sigma_W) = (0.8, 0.5, 1.0, 1.0)$.
- We compute the score vector using Fisher's identity

$$\nabla \log p_\theta(y_{1:n}) = \int \nabla \log p_\theta(x_{1:n}, y_{1:n}) \;\; p_\theta(x_{1:n} | y_{1:n}) \, dx_{1:n}$$

at the true value of θ and compare to its true value.

Empirical Variance for Standard vs FFBS Approximations



Standard path-based (left) vs FFBS (right); the vertical scale is different

Parameter Estimation using Gradient Ascent/EM

- *Gradient ascent*: To maximise $p_\theta(y_{1:n})$ w.r.t θ, use at iteration $k+1$

$$\theta_{k+1} = \theta_k + \nabla \log p_\theta \left(y_{1:n} \right) \big|_{\theta=\theta_k}$$

where $\nabla \log p_\theta \left(y_{1:n} \right) \big|_{\theta=\theta_k}$ is computed using Fisher's identity or IPA (Coquelin, Deguest & Munos, 2009) and any SMC smoothing algorithm.

Parameter Estimation using Gradient Ascent/EM

- *Gradient ascent*: To maximise $p_\theta(y_{1:n})$ w.r.t θ, use at iteration $k+1$

$$\theta_{k+1} = \theta_k + \nabla \log p_\theta \left(y_{1:n} \right)\big|_{\theta=\theta_k}$$

  where $\nabla \log p_\theta \left(y_{1:n} \right)\big|_{\theta=\theta_k}$ is computed using Fisher's identity or IPA (Coquelin, Deguest & Munos, 2009) and any SMC smoothing algorithm.

- *EM algorithm*: To maximise $p_\theta(y_{1:n})$ w.r.t θ, the EM uses at iteration $k+1$

$$\theta_{k+1} = \arg\max \; Q(\theta_k, \theta).$$

  where

$$Q(\theta_k, \theta) = \int \log p_\theta(x_{1:n}, y_{1:n}) \; p_{\theta_k}(x_{1:n} | y_{1:n}) dx_{1:n}$$

  can be computed using any SMC smoothing algorithm.

Online Parameter Estimation using Gradient Ascent/EM

- In the online implementation (Le Gland & Mevel, 1997), update the parameter at time $n+1$ using

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$$

where $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$ and

$$\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1}) = \nabla \log p_{\theta_{1:n}}(y_{1:n}) - \nabla \log p_{\theta_{1:n-1}}(y_{1:n-1}).$$

Online Parameter Estimation using Gradient Ascent/EM

- In the online implementation (Le Gland & Mevel, 1997), update the parameter at time $n+1$ using

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$$

where $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$ and

$$\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1}) = \nabla \log p_{\theta_{1:n}}(y_{1:n}) - \nabla \log p_{\theta_{1:n-1}}(y_{1:n-1}).$$

- An estimate of $\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$ with a time-uniform bounded variance can be computed using online SMC FFBS estimate (Del Moral, D. & Singh, 2009).

Online Parameter Estimation using Gradient Ascent/EM

- In the online implementation (Le Gland & Mevel, 1997), update the parameter at time $n+1$ using

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$$

where $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$ and

$$\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1}) = \nabla \log p_{\theta_{1:n}}(y_{1:n}) - \nabla \log p_{\theta_{1:n-1}}(y_{1:n-1}).$$

- An estimate of $\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$ with a time-uniform bounded variance can be computed using online SMC FFBS estimate (Del Moral, D. & Singh, 2009).
- A numerically stable SMC implementation of online EM (e.g. Cappé, 2009; Elliott, Ford & Moore, 2002) can also be implemented using online SMC FFBS estimate.

Online Parameter Estimation using Gradient Ascent/EM

- In the online implementation (Le Gland & Mevel, 1997), update the parameter at time $n+1$ using

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$$

where $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$ and

$$\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1}) = \nabla \log p_{\theta_{1:n}}(y_{1:n}) - \nabla \log p_{\theta_{1:n-1}}(y_{1:n-1}).$$

- An estimate of $\nabla \log p_{\theta_{1:n}}(y_n | y_{1:n-1})$ with a time-uniform bounded variance can be computed using online SMC FFBS estimate (Del Moral, D. & Singh, 2009).
- A numerically stable SMC implementation of online EM (e.g. Cappé, 2009; Elliott, Ford & Moore, 2002) can also be implemented using online SMC FFBS estimate.
- These non-Bayesian procedures do not suffer from the degeneracy problem but require long data sets for convergence.