SMC for Recursive Parameter Estimation in State-Space Models

Arnaud Doucet
Departments of Statistics & Computer Science
University of British Columbia

Problem Statement

- $\{X_n\}_{n\geq 1}$ latent/hidden Markov process with

$$X_1 \sim \mu_\theta(\cdot) \text{ and } X_n | (X_{n-1} = x) \sim f_\theta(\cdot | x).$$

- $\{Y_n\}_{n\geq 1}$ observation process such that observations are conditionally independent given $\{X_n\}_{n\geq 1}$ and

$$Y_n | (X_n = x) \sim g_\theta(\cdot | x).$$

- **Objectives**: Assume the observations available correspond to $\theta = \theta^*$, obtain a recursive algorithm to estimate θ^*.

Examples

- *Linear Gaussian state-space model*

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \ X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. In this case, we have $\theta = (\alpha, \sigma_v, \sigma_w)$.

- *Stochastic volatility model*

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \ X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= \beta \exp(X_n/2) W_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. In this case, we have $\theta = (\alpha, \sigma_v, \beta)$.

Approaches to Recursive Parameter Estimation

- Bayesian approaches where θ is an unknown random parameter with a prior $p(\theta)$. In this case, inference relies on the sequence of distributions $p(\theta|\, y_{1:n})$.
- Point estimation based on recursive Maximum Likelihood and pseudo-likelihood approaches.

Bayesian Approaches

- In a Bayesian framework, θ is an unknown random parameter with a prior $p(\theta)$.
- At time n, inference relies on

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

where

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

- We know the sequence of distributions $p(x_{1:n}, \theta | y_{1:n})$ up to a normalizing constant so we can use SMC methods.

Preliminary Warning

- We have

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

- We have seen previously that, even for a fixed value θ, the SMC estimate $\widehat{p}(\, y_{1:n}|\, \theta)$ of $p(\, y_{1:n}|\, \theta)$ is under favourable mixing assumptions such that

$$\frac{\mathbb{V}\left[\widehat{p}(\, y_{1:n}|\, \theta)\right]}{p(\, y_{1:n}|\, \theta)^2} \leq C\frac{n}{N};$$

  i.e. the performance degrade linearly with the time index n.

- Intuitively, estimating the whole posterior $p(\theta|\, y_{1:n})$ is obviously more difficult that estimating $p(\, y_{1:n}|\, \theta)$ for a specific value of θ. Hence the SMC algorithms targetting $p(\theta|\, y_{1:n})$ might not enjoy very good convergence properties... Indeed this is unfortunately the case.

SMC Approximations

- Numerous SMC schemes have been proposed to address this problem.
- I will only discuss schemes providing asymptotically consistent estimates of $p\left(x_{1:n}, \theta \mid y_{1:n}\right)$, hence of $p\left(\theta \mid y_{1:n}\right)$; i.e. for n fixed we have convergence for $N \rightarrow \infty$.
- Approaches introducing some artificial random walk dynamics on the parameter/making fixed-lag approximations do not satisfy this property.

Naive SMC Scheme for Parameter Estimation

- Sample $\left(X_1^{(i)}, \theta_0^{(i)}\right) \sim q\left(\cdot, \cdot \mid y_1\right)$ and
$$W_1^{(i)} \propto \frac{p\left(\theta_0^{(i)}\right)\mu_{\theta_0^{(i)}}\left(X_1^{(i)}\right)g_{\theta_0^{(i)}}\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)}, \theta_0^{(i)} \middle| y_1\right)}.$$
- Resample $\left\{\left(X_1^{(i)}, \theta_0^{(i)}\right), W_1^{(i)}\right\}$ to obtain particles $\left\{X_1^{(i)}, \theta_1^{(i)}\right\}$
- At time $n \geq 2$, sample $X_n^{(i)} \sim q_{\theta_{n-1}^{(i)}}\left(\cdot \mid y_n, X_{n-1}^{(i)}\right)$ and
$$W_n^{(i)} \propto \frac{f_{\theta_{n-1}^{(i)}}\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right)g_{\theta_{n-1}^{(i)}}\left(y_n | X_n^{(i)}\right)}{q_{\theta_{n-1}^{(i)}}\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)}.$$
- Resample $\left\{\left(X_{1:n}^{(i)}, \theta_{n-1}^{(i)}\right), W_n^{(i)}\right\}$ to obtain particles $\left\{X_{1:n}^{(i)}, \theta_n^{(i)}\right\}$

- This is just a standard SMC scheme...
- We have

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

- In particular, we have

$$\widehat{p}\left(\theta \middle| y_{1:n} \right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\theta_n^{(i)}} \left(\theta \right)$$

where $\theta_n^{(i)}$ correspond to the particles having been sampled at time 1 which have survived to the resampling steps at time $1, 2, \ldots, n$.

Performance

- This algorithm provides an asymptotically consistent estimate of the targets under very weak assumptions....
- ... and yes it is a very bad algorithm. We only sample particles in the Θ space at time 1; this is followed by successive resampling steps.
- After a few time steps, we have

$$\widehat{p}\left(\theta \mid y_{1:n}\right) = \delta_{\overline{\theta}}\left(\theta\right)$$

  where $\theta_n^{(i)} = \overline{\theta}$ for $i \in \{1, \ldots, N\}$. This is somewhat similar to the problem we faced before when there was no unknown parameter but we were interested in estimating $p\left(x_1 \mid y_{1:n}\right)$... but the problem is even worse as, because of the lack of ergodicity, this error propagate itself.
- Theoretically, it means that we do not have a uniform convergence result for $\widehat{p}\left(\theta \mid y_{1:n}\right)$; only the following very weak result

$$\mathbb{E}\left[\left|\int \varphi\left(\theta\right)\left(\widehat{p}\left(d\theta \mid y_{1:n}\right) - p\left(d\theta \mid y_{1:n}\right)\right)\right|^p\right]^{1/p} \leq \frac{c\left(n\right)}{\sqrt{N}}$$

  where $c\left(n\right)$ increases over time.

How to improve performance?

- We can use all the advanced methods discussed previously: auxiliary method, resample-move, block sampling.
- Resample move is especially attractive in this context: it consists in adding at time n an MCMC move $K_n\left(x'_{1:n}, \theta' \middle| x_{1:n}, \theta\right)$ of invariant distribution $p\left(x_{1:n}, \theta \middle| y_{1:n}\right)$. To keep the algorithm on-line, we can only update a fixed number of variables; say here θ only.
- For example, we could use a Gibbs step

$$K_n\left(x'_{1:n}, \theta' \middle| x_{1:n}, \theta\right) = \delta_{x_{1:n}}\left(x'_{1:n}\right) p\left(\theta' \middle| y_{1:n}, x_{1:n}\right)$$

Resample Move SMC for Parameter Estimation

- Sample $\left(X_1^{(i)}, \theta_0^{(i)}\right) \sim q\left(\cdot, \cdot \mid y_1\right)$ and
$$W_1^{(i)} \propto \frac{p\left(\theta_0^{(i)}\right)\mu_{\theta_0^{(i)}}\left(X_1^{(i)}\right)g_{\theta_0^{(i)}}\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)}, \theta_0^{(i)} \middle| y_1\right)}.$$
- Resample $\left\{\left(X_1^{(i)}, \theta_0^{(i)}\right), W_1^{(i)}\right\}$ to obtain particles $\left\{X_1^{(i)}, \overline{\theta}_1^{(i)}\right\}$.
- Sample $\theta_1^{(i)} \sim p\left(\cdot \mid y_1, X_1^{(i)}\right)$.
- At time $n \geq 2$, sample $X_n^{(i)} \sim q_{\theta_{n-1}^{(i)}}\left(\cdot \mid y_n, X_{n-1}^{(i)}\right)$ and
$$W_n^{(i)} \propto \frac{f_{\theta_{n-1}^{(i)}}\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right)g_{\theta_{n-1}^{(i)}}\left(y_n | X_n^{(i)}\right)}{q_{\theta_{n-1}^{(i)}}\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)}.$$
- Resample $\left\{\left(X_{1:n}^{(i)}, \theta_{n-1}^{(i)}\right), W_n^{(i)}\right\}$ to obtain particles $\left\{X_{1:n}^{(i)}, \overline{\theta}_n^{(i)}\right\}$.
- Sample $\theta_n^{(i)} \sim p\left(\cdot \mid y_{1:n}, X_{1:n}^{(i)}\right)$.

Implementation Issues

- At first glance, this algorithm seems difficult to implement as it requires storing the paths $\left\{ X_{1:n}^{(i)} \right\}$ so memory requirements increase.
- However, in many practical applications, we have

$$p\left(\theta \mid y_{1:n}, x_{1:n}\right) = p\left(\theta \mid s_n\left(x_{1:n}, y_{1:n}\right)\right)$$

i.e. it depends only on a set of sufficient statistics $s_n\left(x_{1:n}, y_{1:n}\right)$ of fixed dimension.

Example: Linear Gaussian state-space model

- We have

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \; X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

  where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.
- Assume for sake of simplicity that only α is unknown with $p(\alpha) = \mathcal{U}_{[-1,1]}(\alpha)$.
- It is easy to check that

$$p(\alpha | y_{1:n}, x_{1:n}) \propto \mathcal{N}(\alpha; m_n, \sigma_n^2) 1_{[-1,1]}(\alpha)$$

  where

$$\sigma_n^2 = \left(\sum_{k=1}^{n-1} x_k^2 \right)^{-1}, \; m_n = \sigma_n^2 \left(\sum_{k=2}^{n} x_{k-1} x_k \right).$$

- In practice, we only need to store $\sum_{k=2}^{n} x_{k-1} x_k$ and $\sum_{k=1}^{n-1} x_k^2$ instead of $x_{1:n}$.

Resample Move SMC with Sufficient Statistics for Parameter Estimation

- $\left(X_1^{(i)}, \theta_0^{(i)}\right) \sim q\left(\cdot, \cdot | \, y_1\right)$ and $W_1^{(i)} \propto \frac{p\left(\theta_0^{(i)}\right)\mu_{\theta_0^{(i)}}\left(X_1^{(i)}\right)g_{\theta_0^{(i)}}\left(y_1|X_1^{(i)}\right)}{q\left(X_1^{(i)},\theta_0^{(i)}\middle|y_1\right)}$.
- Resample $\left\{\left(X_1^{(i)}, \theta_0^{(i)}\right), W_1^{(i)}\right\}$ to obtain $\left\{X_1^{(i)}, s_1\left(X_1^{(i)}, y_1\right), \overline{\theta}_1^{(i)}\right\}$.
- $\theta_1^{(i)} \sim p\left(\cdot | \, s_1\left(X_1^{(i)}, y_1\right)\right)$.
- At time $n \geq 2$, $X_n^{(i)} \sim q_{\theta_{n-1}^{(i)}}\left(\cdot | \, y_n, X_{n-1}^{(i)}\right)$ and
$$W_n^{(i)} \propto \frac{f_{\theta_{n-1}^{(i)}}\left(X_n^{(i)}\middle|X_{n-1}^{(i)}\right)g_{\theta_{n-1}^{(i)}}\left(y_n|X_n^{(i)}\right)}{q_{\theta_{n-1}^{(i)}}\left(X_n^{(i)}\middle|y_n,X_{n-1}^{(i)}\right)}.$$
- Resample $\left\{\left(X_n^{(i)}, s_n\left(X_{1:n}^{(i)}, y_{1:n}\right), \theta_{n-1}^{(i)}\right), W_n^{(i)}\right\}$ to obtain $\left\{X_n^{(i)}, s_n\left(X_{1:n}^{(i)}, y_{1:n}\right), \overline{\theta}_n^{(i)}\right\}$.
- Sample $\theta_n^{(i)} \sim p\left(\cdot | \, s_n\left(X_{1:n}^{(i)}, y_{1:n}\right)\right)$.

Comments

- This algorithm appears elegant.
- This algorithm and some variations have already appeared several times in the literature (Andrieu, De Freitas & D., 1999), (Fearnhead, 2002), (Storvik, 2002), (Johannes & Polson, 2007).
- This algorithm suffers from very severe limitations and is not robust as, once more, it relies implicitly on the SMC approximation of a sequence of distributions $p\left(x_{1:n} \middle| y_{1:n}\right)$ of increasing dimension; the pitfalls of this approach were first discussed in (Andrieu, De Freitas & D., 1999), see also (Andrieu, D. & Tadic, 2005).

Illustration of the degeneracy phenomenon

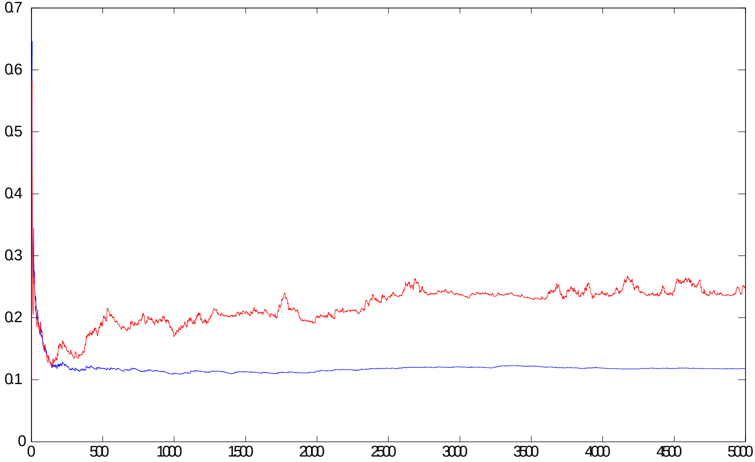

Figure: Sufficient statistics computed exactly through the Kalman smoother (blue) and the SMC method (red).

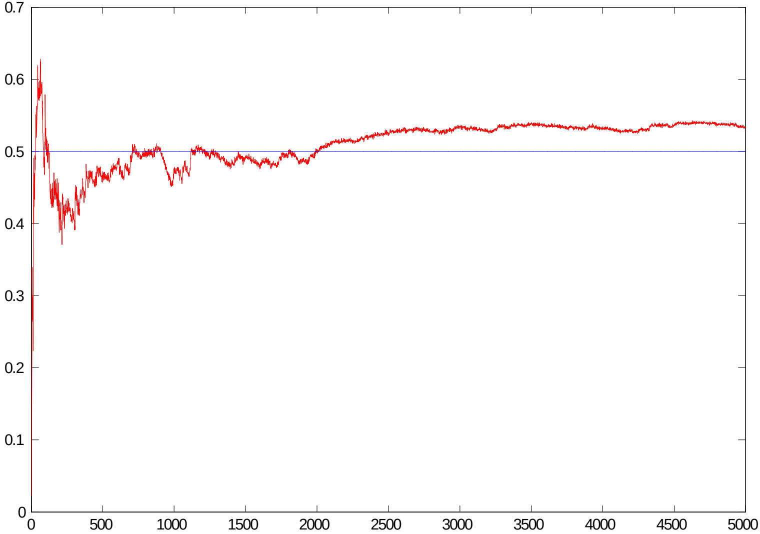

Figure: SMC approximation of $\mathbb{E}\left[\theta \middle| y_{1:n}\right]$ for $N = 1000$ particles (red) as a function of n versus true value (blue). The algorithm converges towards a wrong value.

Additional Comments

- These algorithms provide asymptotically consistent approximations; i.e. for fixed n, the SMC approximation converges towards the true target as N increases...
- Still, it does not mean that such algorithms perform well in practice. For a fixed N and an increasing n, the error will increase; i.e. it is not possible to obtain uniform convergence results.
- You can use any advanced method you want but, as long as you rely on an SMC approximation of $p\left(x_{1:n} \middle| y_{1:n}\right)$ (or $p\left(s_n\left(x_{1:n}, y_{1:n}\right) \middle| y_{1:n}\right)$), then you will face the same problem eventually for n large enough.
- For a **fixed time horizon**, and N large enough, such methods might perform reasonably well and cannot be completely ruled out. However you have to be extremely careful: determining a large enough N is difficult (see SMC project for more information and quantitative results).

- The credible intervals estimates computed via such approaches are much tighter than they should be (because of the degeneracy phenomenon) so you cannot trust them.
- You can expect these methods to perform very poorly when the dimension of the parameter space is high; say superior to 5-10.

Discussion and Future Work

- It is impossible to obtain an asymptotically convergent SMC algorithm to estimate $p(\theta | y_{1:n})$ which enjoys uniform convergence properties.
- At the price of a non-vanishing bias, it should be possible to obtain much better approximations of $p(\theta | y_{1:n})$ based on fixed-lag approximations. The main problem is that it is difficult to quantify the bias in practical situations.

Recursive Maximum Likelihood

- Recursive Maximum Likelihood is a fairly old and popular approach in the system identification/control community.
- We show here how to implement an SMC version of it for general state-space models.
- Under stationary assumptions (e.g. Tadic & D., 2005), we have

$$\frac{1}{n}\log p_\theta\left(Y_{1:n}\right)=\frac{1}{n}\sum_{k=1}^{n}\log p_\theta\left(\left.Y_k\right|Y_{1:k-1}\right)\rightarrow l\left(\theta\right)$$

with

$$l\left(\theta\right)=\int\int_{\mathcal{Y}\times\mathcal{P}(\mathcal{X})}\log\left(\int g_\theta\left(\left.y\right|x\right)\mu\left(x\right)dx\right)\lambda_{\theta,\theta^*}\left(dy,d\mu\right),$$

where $\mathcal{P}\left(\mathcal{X}\right)$ is the space of probability distributions on $\mathcal{X}$ and $\lambda_{\theta,\theta^*}\left(dy,d\mu\right)=\int\lambda_{\theta,\theta^*}\left(dx,dy,d\mu\right)$; $\lambda_{\theta,\theta^*}\left(dx,dy,d\mu\right)$ being the invariant distribution of the Markov chain $\left\{X_n,Y_n,p_\theta\left(\left.x_n\right|Y_{1:n-1}\right)\right\}_{n\geq 1}$.

Stochastic Approximation

- The set of global maxima of the averaged log-likelihood $l(\theta)$ includes θ^*.
- The function $l(\theta)$ is unknown but can be maximized using a stochastic approximation algorithm

$$\theta_n = \theta_{n-1} + \gamma_n \nabla \log\ p_{\theta_{1:n-1}}\left(Y_n \middle| Y_{1:n-1}\right) \tag{1}$$

  where the stepsize sequence $\{\gamma_n\}_{n\geq 1}$ is a positive non-increasing sequence.
- $p_{\theta_{1:n}}\left(x_n \middle| Y_{1:n}\right)$ denotes the filter computed using θ_{t-1} at time t and similarly for $\nabla \log\ p_{\theta_{1:n-1}}\left(Y_n \middle| Y_{1:n-1}\right)$.
- We typically need $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$; i.e. one selects $\gamma_n = \gamma_0.n^{-\alpha}$ where $\gamma_0 > 0$ and $0.5 < \alpha \leq 1$.
- This algorithm is a stochastic gradient algorithm and is not guaranteed to converge towards θ^*; only to a local maximum of $l(\theta)$.
- For finite-state space hidden Markov models, this algorithm was proposed and studied by (Le Gland & Mevel, 1997).

SMC Approximation

- We need to approximate $\nabla \log p_\theta (Y_n | Y_{1:n-1})$.
- The first approach consists of using

$$\nabla \log p_\theta (Y_n | Y_{1:n-1}) = \nabla \log p_\theta (Y_{1:n}) - \nabla \log p_\theta (Y_{1:n-1})$$

where Fisher's identity yields

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

with

$$\begin{aligned} \nabla \log p_\theta (x_{1:n}, Y_{1:n}) &= \nabla \log \mu_\theta (x_1) + \nabla \log g_\theta (Y_1 | x_1) \\ &\quad + \sum_{k=2}^{n} \nabla \log f_\theta (x_k | x_{k-1}) + \nabla \log g_\theta (Y_k | x_k) . \end{aligned}$$

SMC Implementation of Fisher's identity

- Given you favourite SMC approximation $\widehat{p}_\theta\left(\left.x_{1:n}\right| Y_{1:n}\right)$ of $p_\theta\left(\left.x_{1:n}\right| Y_{1:n}\right)$; say

$$\widehat{p}_\theta\left(\left.x_{1:n}\right| Y_{1:n}\right)=\sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$$

  then we can compute an estimate

$$\widehat{\nabla \log p_\theta}\left(Y_{1:n}\right)=\sum_{i=1}^{N} W_n^{(i)} \nabla \log p_\theta\left(X_{1:n}^{(i)}, Y_{1:n}\right).$$

- This estimate can be easily computed recursively using

$$\begin{gathered}\nabla \log p_\theta\left(x_{1:n}, Y_{1:n}\right)=\nabla \log p_\theta\left(x_{1:n-1}, Y_{1:n-1}\right) \\ +\nabla \log f_\theta\left(\left.x_n\right| x_{n-1}\right)+\nabla \log g_\theta\left(\left.Y_n\right| x_n\right).\end{gathered}$$

- We obtain $\widehat{\nabla \log p_\theta}\left(\left.Y_n\right| Y_{1:n-1}\right)=\widehat{\nabla \log p_\theta}\left(Y_{1:n}\right)-\widehat{\nabla \log p_\theta}\left(Y_{1:n-1}\right)$ but this estimate has poor properties as, once more, it relies implicitly on an approximation of the joint distribution $p_\theta\left(\left.x_{1:n}\right| Y_{1:n}\right)$...

SMC Approximation of the Sensitivity Equations

- There is an alternative way to compute $\nabla \log p_\theta \left(Y_n \middle| Y_{1:n-1} \right)$ based on sensitivity equations.
- We have

$$\nabla \log p_\theta \left(Y_{n+1} \middle| Y_{1:n} \right) = \int \nabla \log p_\theta \left(x_{n+1}, Y_{n+1} \middle| Y_{1:n} \right) \; p \left(x_{n+1} \middle| Y_{1:n+1} \right) dx_{n+1}$$

with

$$\nabla p_\theta \left(x_{n+1}, Y_{n+1} \middle| Y_{1:n} \right) = g_\theta \left(Y_{n+1} \middle| x_{n+1} \right) \int f_\theta \left(x_{n+1} \middle| x_n \right) p_\theta \left(x_n \middle| Y_{1:n} \right)$$
$$\times \left(\nabla \log g_\theta \left(Y_{n+1} \middle| x_{n+1} \right) + \nabla \log f_\theta \left(x_{n+1} \middle| x_n \right) + \nabla \log p_\theta \left(x_n \middle| Y_{1:n} \right) \right) dx_n .$$

- By differentiating $\nabla \log p_\theta \left(x_{n+1} \middle| Y_{1:n+1} \right)$, we obtain

$$\nabla p_\theta \left(x_{n+1} \middle| Y_{1:n+1} \right) = \frac{\nabla p_\theta \left(x_{n+1}, Y_{n+1} \middle| Y_{1:n} \right)}{p_\theta \left(Y_{n+1} \middle| Y_{1:n} \right)}$$
$$- p_\theta \left(x_{n+1} \middle| Y_{1:n+1} \right) \nabla \log p_\theta \left(Y_{n+1} \middle| Y_{1:n} \right)$$

SMC Approximation of Filter Sensitivity

- To implement this recursion, we need to approximate $\nabla p_{\theta}\left(x_n \middle| Y_{1:n}\right)$. This is a signed measure such that

$$\int \nabla p_{\theta}\left(x_n \middle| Y_{1:n}\right) dx_n = 0.$$

- A first idea to approximate $\nabla p_{\theta}\left(x_n \middle| Y_{1:n}\right)$ consists of using the identity

$$\nabla p_{\theta}\left(x_n \middle| Y_{1:n}\right) = \int \nabla \log p_{\theta}\left(x_{1:n} \middle| Y_{1:n}\right) . p_{\theta}\left(x_{1:n} \middle| Y_{1:n}\right) dx;$$

  this would rely once more on an SMC approximation of $p_{\theta}\left(x_{1:n} \middle| Y_{1:n}\right)$... and it is just a convoluted way to rewrite the previous algorithm.

- An alternative consists of using (Poyadjis, D. & Singh, 2005)

$$\nabla p_\theta\left(x_n \middle| Y_{1:n}\right) = \frac{\nabla p_\theta\left(x_n \middle| Y_{1:n}\right)}{p_\theta\left(x_n \middle| Y_{1:n}\right)} . p_\theta\left(x_n \middle| Y_{1:n}\right);$$

that is if $\widehat{p}_\theta\left(x_n \middle| Y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}\left(x_n\right)$ then

$$\widehat{\nabla p}_\theta\left(x_n \middle| Y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \frac{\widetilde{\nabla p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)}{\widetilde{p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)} \delta_{X_n^{(i)}}\left(x_n\right)$$

- This only relies on approximation of the marginals; the price to pay is that we now need a pointwise estimate of $\widetilde{p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)$ and $\widetilde{\nabla p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)$. The algorithm is thus in $O\left(N^2\right)$.

SMC Approximation

- At time $n-1$, assume approximations of the filtering distribution and its derivatives of the form

$$\begin{aligned}\widehat{p}_\theta\left(x_{n-1}\middle|\, y_{1:n-1}\right) &= \sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{X_{n-1}^{(i)}}\left(x_{n-1}\right), \\ \widehat{\nabla p}_\theta\left(x_{n-1}\middle|\, y_{1:n-1}\right) &= \sum_{i=1}^{N} W_{n-1}^{(i)} A_{n-1}^{(i)} \delta_{X_{n-1}^{(i)}}\left(x_{n-1}\right),\end{aligned}$$

  are available where $A_{n-1}^{(i)}$ is an approximation of $\nabla p_\theta\left(X_{n-1}^{(i)}\middle|\, y_{1:n-1}\right) / p_\theta\left(X_{n-1}^{(i)}\middle|\, y_{1:n-1}\right)$.
- We obtain the pointwise approximations of $p_\theta\left(x_n, y_n\middle|\, y_{1:n-1}\right)$, $\nabla p_\theta\left(x_n, y_n\middle|\, y_{1:n-1}\right)$

$$\begin{aligned}\widetilde{p}_\theta\left(x_n, y_n\middle|\, y_{1:n-1}\right) = \textstyle\sum_{i=1}^{N} W_{n-1}^{(i)} g\left(y_n\middle|\, x_n\right) f\left(x_n\middle|\, y_n, X_{n-1}^{(i)}\right), \\ \widetilde{\nabla p}_\theta\left(x_n, y_n\middle|\, y_{1:n-1}\right) = g_\theta\left(y_n\middle|\, x_n\right) \textstyle\sum_{i=1}^{N} W_{n-1}^{(i)} f_\theta\left(x_n\middle|\, X_{n-1}^{(i)}\right) \\ \times\left(\nabla \log g_\theta\left(y_n\middle|\, x_n\right) + \nabla \log f_\theta\left(x_n\middle|\, X_{n-1}^{(i)}\right) + A_{n-1}^{(i)}\right).\end{aligned}$$

- We use a marginalized version of the APF which relies on a joint probability density

$$q_\theta\left(x_n, y_n \mid x_{n-1}\right) = q_\theta\left(x_n \mid y_n, x_{n-1}\right) q_\theta\left(y_n \mid x_{n-1}\right)$$

which is an approximation of

$$p_\theta\left(x_n, y_n \mid x_{n-1}\right) = g_\theta\left(y_n \mid x_n\right) f_\theta\left(x_n \mid x_{n-1}\right)$$

- We construct the marginal importance distribution

$$\begin{aligned} q_\theta\left(x_n \mid y_n\right) &= \sum_{i=1}^{N} \widetilde{W}_n^{(i)} q_n\left(x_n \mid y_n, X_{n-1}^{(i)}\right), \\ \widetilde{W}_n^{(i)} &\propto W_{n-1}^{(i)} q_\theta\left(y_n \mid X_{n-1}^{(i)}\right). \end{aligned}$$

- Sampling from $q_\theta\left(x_n \mid y_n\right)$ includes implicitly the resampling step.

SMC for Sensitivity

- Sample $X_n^{(i)} \sim q_\theta\left(\cdot \middle| y_n\right)$.
- Evaluate

$$w_n^{(i)} = \frac{\widetilde{p}_\theta\left(X_n^{(i)}, y_n \middle| y_{1:n-1}\right)}{q_\theta\left(X_n^{(i)} \middle| y_n\right)}, \quad a_n^{(i)} = \frac{\widetilde{\nabla p_\theta}\left(X_n^{(i)}, y_n \middle| y_{1:n-1}\right)}{q_\theta\left(X_n^{(i)} \middle| y_n\right)}$$

$$\begin{aligned} W_n^{(i)} &\propto w_n^{(i)} \text{ with } \sum_{i=1}^{N} W_n^{(i)} = 1, \\ W_n^{(i)} A_n^{(i)} &= \frac{a_n^{(i)}}{\sum_{j=1}^{N} w_n^{(i)}} - W_n^{(i)} \frac{\sum_{j=1}^{N} a_n^{(j)}}{\sum_{j=1}^{N} w_n^{(j)}}, \end{aligned}$$

- We have

$$\widehat{\nabla \log p_\theta}\left(Y_n \middle| Y_{1:n-1}\right) = \frac{\sum_{i=1}^{N} a_n^{(i)}}{\sum_{i=1}^{N} w_n^{(i)}}.$$

SMC for Recursive Maximum Likelihood

- Sample $X_n^{(i)} \sim q_{\theta_{n-1}}\left(\cdot \middle| y_n\right)$.
- Evaluate

$$w_n^{(i)} = \frac{\widetilde{p}_{\theta_{n-1}}\left(X_n^{(i)}, y_n \middle| y_{1:n-1}\right)}{q_{\theta_{n-1}}\left(X_n^{(i)} \middle| y_n\right)}, \; a_n^{(i)} = \frac{\widetilde{\nabla p}_{\theta_{n-1}}\left(X_n^{(i)}, y_n \middle| y_{1:n-1}\right)}{q_{\theta_{n-1}}\left(X_n^{(i)} \middle| y_n\right)}$$

$$\begin{aligned} W_n^{(i)} &\propto w_n^{(i)} \text{ with } \sum_{i=1}^{N} W_n^{(i)} = 1, \\ W_n^{(i)} A_n^{(i)} &= \frac{a_n^{(i)}}{\sum_{j=1}^{N} w_n^{(i)}} - W_n^{(i)} \frac{\sum_{j=1}^{N} a_n^{(j)}}{\sum_{j=1}^{N} w_n^{(j)}}, \end{aligned}$$

- Update the parameter

$$\theta_n = \theta_{n-1} + \gamma_n \frac{\sum_{i=1}^{N} a_n^{(i)}}{\sum_{i=1}^{N} w_n^{(i)}}.$$

Comments

- This algorithm is perhaps not very elegant but simple.
- This algorithm only relies on the SMC approximation of the marginals $p\left(x_n \middle| y_{1:n} \right)$.
- Under standard mixing assumptions, we can establish uniform convergence results for $\widehat{\nabla p_\theta}\left(x_n \middle| y_{1:n} \right)$.
- There is no accumulation of errors over time contrary to the SMC approaches discussed earlier.
- It has been used successfully for high-dimensional parameter estimation problems arising in robotics and bioinformatics.
- The observed information matrix can be computed similalry.

Limitations of this approach

- It is in $O\left(N^2\right)$ although fast methods can be used to speed it up.
- It requires scaling the step-size sequence appropriately for multidimensional parameters.
- It is only useful for large datasets.

Alternative to Stochastic Gradient

- In a batch context, the EM algorithm is a very popular alternative to gradient-type approaches.
- It is possible to derive an online version of the EM.
- However, once more, this algorithm would rely on an SMC approximation of $p_\theta\left(x_{1:n}\middle|\,y_{1:n}\right)$.
- A simple fixed-lag approximation can be used to mitigate this problem but not asymptotically consistent (good course project though).

Pseudo-likelihood Approaches

- Instead of trying to maximize the likelihood, we introduce a pseudo-likelihood.
- Assuming a stationary state-space model, we have

$$p_\theta\left(\mathrm{x}_k, \mathrm{y}_k\right) = \pi_\theta(x_{kL+1}) g_\theta(y_{kL+1}|x_{kL+1}) \prod_{i=kL+2}^{(k+1)L} f_\theta(x_i|x_{i-1}) g_\theta(y_i|x_i) \ .$$

- The log pseudo-likelihood for m blocks of observations is given by

$$l_L(\theta, \mathrm{Y}_{0:m-1}) := \sum_{k=0}^{m-1} \log \ p_\theta\left(\mathrm{Y}_k\right) \ , \tag{2}$$

- Compared to the true likelihood, essentially ignores the dependence between data blocks.

- Under ergodicity assumptions, we have

$$\lim_{m\to\infty} \frac{1}{m} l_L(\theta, \mathrm{Y}_{0:m-1}) =: l_L(\theta) \ ,$$

where

$$l_L(\theta) := \int_{\mathrm{Y}^L} \log\left(p_\theta(\mathrm{y})\right) p_{\theta^*}(\mathrm{y}) d\mathrm{y}.$$

- It can be shown that the set of parameters maximizing $l_L(\theta)$ includes the true parameter. This follows from the fact that maximizing $l_L(\theta)$ is equivalent to minimizing the following Kullback-Leibler divergence

$$K_L(\theta, \theta^*) = l_L(\theta^*) - l_L(\theta) \geq 0 \ .$$

On-line EM algorithm

- To introduce the on-line EM, we first present an "ideal" batch EM algorithm to minimize $K_L(\theta, \theta^*)$ with respect to θ or equivalently to maximize $l_L(\theta)$.
- At iteration $k+1$, given an estimate θ_k of θ^*, we update our estimate via

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

where

$$Q(\theta, \theta_k) = \int_{\mathsf{X}^L \times \mathsf{Y}^L} \log(p_\theta(\mathsf{x}, \mathsf{y}))\, p_{\theta_k}(\mathsf{x}|\mathsf{y})\, p_{\theta^*}(\mathsf{y}) d\mathsf{x} d\mathsf{y} \;.$$

- Now for any $\theta \in \Theta$

$$\begin{aligned} Q(\theta_{k+1}, \theta_k) - Q(\theta_k, \theta_k) &= K_L(\theta_k, \theta^*) - K_L(\theta_{k+1}, \theta^*) \\ &+ \int_{\mathsf{X}^L \times \mathsf{Y}^L} \log\left(\frac{p_{\theta_{k+1}}(\mathsf{x}|\mathsf{y})}{p_{\theta_k}(\mathsf{x}|\mathsf{y})}\right) p_{\theta_k}(\mathsf{x}|\mathsf{y})\, p_{\theta^*}(\mathsf{y}) d\mathsf{x} d\mathsf{y} \end{aligned}$$

so an iteration of this "ideal" EM algorithm decreases the value of $K_L(\theta_k, \theta^*)$.

- In practice for the models which we will consider, it is necessary to compute a set of sufficient statistics $\Phi\left(\theta_k, \theta^*\right)$ at time k in order to compute Q.
- In practice, $Q\left(\theta, \theta_{k-1}\right)$ cannot be computed as the expectations appearing in the expression for $\Phi\left(\theta_k, \theta^*\right)$ are with respect to a measure dependent on the unknown parameter value θ^*.
- Thanks to the ergodicity and stationarity assumptions, the observations $\{\mathrm{Y}_k\}$ provide us with samples from $p_{\theta^*}(\mathrm{y})$ which can be used for the purpose of Monte Carlo integration

$$\hat{\Phi}_k = (1-\gamma_k)\,\hat{\Phi}_{k-1} + \gamma_k \mathbb{E}_{\theta_{k-1}}\left(\Psi\left(\mathrm{X}_k, \mathrm{Y}_k\right) \middle| \mathrm{Y}_k \right), \tag{3}$$

where $\mathbb{E}_{\theta_{k-1}}\left(\phi(X_k) \middle| \mathrm{Y}_k\right)$ denotes the expectation of ϕ with respect to $p_{\theta_{k-1}}\left(\mathrm{x}_k \middle| \mathrm{Y}_k\right)$.

- We then substitute $\hat{\Phi}_k$ for $\Phi\left(\theta_k, \theta^*\right)$ and obtain $\theta_k = \Lambda(\hat{\Phi}_k)$.
- If θ_k was constant and $\gamma_k = k^{-1}$ then $\hat{\Phi}_k$ would simply compute the arithmetic average of $\left\{\mathbb{E}_{\theta_{k-1}}\left(\Psi\left(\mathrm{X}_k, \mathrm{Y}_k\right) \middle| \mathrm{Y}_k \right)\right\}$, and converge towards $\Phi\left(\theta_k, \theta^*\right)$ by ergodicity.

- To summarize, the vector of sufficient statistics $\hat{\Phi}_{-1}$ is arbitrarily initialized and the on-line EM algorithm proceeds as follows for the data block indexed by $k \geq 0$.
- *E-step*

$$\hat{\Phi}_k = (1-\gamma_k)\hat{\Phi}_{k-1} + \gamma_k \mathbb{E}_{\theta_{k-1}}\left(\Psi\left(X_k, Y_k\right) \middle| Y_k \right) \ .$$

- *M-step*

$$\theta_k = \Lambda(\hat{\Phi}_k) \ .$$

- In scenarios where $\mathbb{E}_{\theta_k}\left(\Psi\left(X_k, Y_k\right) \middle| Y_k \right)$ does not admit an analytical expression, a further Monte Carlo approximation can be used.

- Assume that a good approximation $q_{\theta_{k-1}}(\mathrm{x}_k|\mathrm{Y}_k)$ of $p_{\theta_{k-1}}(\mathrm{x}_k|\mathrm{Y}_k)$ is available, and that it is easy to sample from $q_{\theta_{k-1}}(\mathrm{x}_k|\mathrm{Y}_k)$.
- *E-step*

$$\begin{aligned} \mathrm{X}_k^{(i)} &\sim q_{\theta_{k-1}}(\cdot|\mathrm{Y}_k) \text{ for } i = 1, \ldots, N , \\ \hat{\Phi}_k &= (1-\gamma_k)\hat{\Phi}_{k-1} + \gamma_k \sum_{i=1}^{N} W_k^{(i)} \Psi(\mathrm{X}_k^{(i)}, \mathrm{Y}_k) , \end{aligned}$$

where

$$W_k^{(i)} \propto \frac{p_{\theta_{k-1}}(\mathrm{X}_k^{(i)}, \mathrm{Y}_k)}{q_{\theta_{k-1}}(\mathrm{X}_k^{(i)}|\mathrm{Y}_k)}, \ \sum_{i=1}^{N} W_k^{(i)} = 1 .$$

- *M-step*

$$\theta_k = \Lambda(\hat{\Phi}_k) .$$

- If it is possible to sample from $p_{\theta_{k-1}}(\mathrm{x}_k|\mathrm{Y}_k)$ exactly then it is not necessary to have a large N, $N = 1$ is sufficient. Indeed it is only necessary to produce estimates of $\mathbb{E}_{\theta_{k-1}}(\Psi(\mathrm{X}_k, \mathrm{Y}_k)|\, \mathrm{Y}_k)$.

- Note that as such the algorithm above leads to asymptotically biased estimates, but that this can be easily corrected by considering instead the following recursion for the estimation of the conditional expectation

$$
\begin{aligned}
\hat{F}_k &= (1-\gamma_k)\hat{F}_{k-1} + \gamma_k \frac{1}{N}\sum_{i=1}^{N} \frac{p_{\theta_{k-1}}(\mathrm{X}_k^{(i)}, \mathrm{Y}_k)}{q_{\theta_{k-1}}(\mathrm{X}_k^{(i)}|\mathrm{Y}_k)} \Psi(\mathrm{X}_k^{(i)}, \mathrm{Y}_k) \;, \\
\hat{N}_k &= (1-\gamma_k)\hat{N}_{k-1} + \gamma_k \frac{1}{N}\sum_{i=1}^{N} \frac{p_{\theta_{k-1}}(\mathrm{X}_k^{(i)}, \mathrm{Y}_k)}{q_{\theta_{k-1}}(\mathrm{X}_k^{(i)}|\mathrm{Y}_k)} \;,
\end{aligned}
$$

  and let $\hat{\Phi}_k = \hat{F}_k / \hat{N}_k$.

- SMC techniques can also be used to approximate this expectation. We stress here on the fact that in the situation where SMC methods are used in this context, the path degeneracy issue is easily dealt with since L is fixed, and very often of small dimension.