Exercises set II
PhD course on Sequential Monte Carlo methods 2021

Linköping University and Uppsala University

August 17, 2021

This document contains exercises to make you familiar with the content of the course. *The exercises in this document are not mandatory, and you do not need to hand in your solutions.* The mandatory assignment is found in a separate document named "Hand-in". We strongly recommend that you carefully work through these exercises before starting with the mandatory assignments.

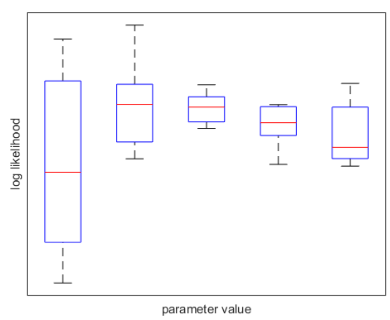
II.1 **Likelihood estimates for the stochastic volatility model**

Consider again (cf. I.4) the stochastic volatility model

$$x_t|x_{t-1} \sim \mathcal{N}(x_t; \phi x_{t-1}, \sigma^2), \tag{1a}$$
$$y_t|x_t \sim \mathcal{N}(y_t; 0, \beta^2 \exp(x_t)), \tag{1b}$$

where the parameter vector is given by $\theta = \{\phi, \sigma, \beta\}$ and the data is found in `seOMXlogreturns2012to2014.csv`.

(a) Let β be unknown, and assume the other parameters are $\phi = 0.98$ and $\sigma = 0.16$. Make a reasonably coarse grid for β between 0 to 2, and implement the bootstrap particle filter to estimate the likelihood for each of these values of β. Run the particle filter 10 times for every parameter combination, and present the result as a box plot similar to this:



For numerical reasons, it is usually better to consider the *log* likelihood, i.e., the logarithm of (10)

$$\log \widehat{p}(y_{1:T}) = \log \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} \underbrace{p(y_t \mid x_t^i)}_{\widetilde{w}_t^i} = \sum_{t=1}^{T} \left(\log \sum_{i=1}^{N} \widetilde{w}_t^i - \log N \right). \tag{2}$$

It is, however, important to realize that $\mathbb{E}\left[\widehat{p}(y_{1:T})\right] = p(y_{1:T})$ does *not* imply $\mathbb{E}\left[\log \widehat{p}(y_{1:T})\right] = \log p(y_{1:T})$!

(b) Study how N and T affects the variance in the log likelihood estimate.

(c) Remove the resampling step from your particle filter algorithm, and study its effect on the variance of the estimator.

II.2 **Fully adapted particle filter**

(a) Motivate for each of these model why it is/is not possible to implement the fully adapted particle filter for it.

(i)

$$x_{t+1} = 0.4x_t + v_t, \qquad v_t \sim \mathcal{N}(0,1), \tag{3}$$
$$y_t = -0.5x_t + e_t \qquad e_t \sim \mathcal{U}([-2,2]). \tag{4}$$

(ii)

$$x_{t+1} = \cos(x_t)^2 + v_t, \qquad v_t \sim \mathcal{N}(0,1), \tag{5}$$
$$y_t = 2x_t + e_t \qquad e_t \sim \mathcal{N}(0,0.01). \tag{6}$$

(iii)

$$x_{t+1} = \cos(x_t + v_t)^2, \qquad v_t \sim \mathcal{N}(0,1), \tag{7}$$
$$y_t = 2x_t + e_t \qquad e_t \sim \mathcal{N}(0,0.01). \tag{8}$$

(b) Implement the fully adapted particle filter for model (ii), and make a simulation study to compare the variance in the estimates of $\mathbb{E}[X_t \mid y_{1:t}]$ to the estimates obtained by a bootstrap particle filter.

II.3 **Likelihood estimator for the APF.**

The particle filter likelihood estimator is given by

$$\widehat{p}(y_{1:T}) = \prod_{t=1}^{T} \left\{ \frac{1}{N} \sum_i \widetilde{w}_t^i \right\} \tag{9}$$

For the bootstrap particle filter, given in Algorithm 1, a sketchy derivation of this estimator can be done as:

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1}) = \prod_{t=1}^{T} \int p(y_t \mid x_t) p(x_t \mid y_{1:t-1}) dx_t \approx \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} \underbrace{p(y_t \mid x_t^i)}_{\widetilde{w}_t^i} \tag{10}$$

where the particles x_t^i sampled at time t in the bootstrap particle filter (before weighting) can be viewed as approximately distributed according to the predictive distribution $p(x_t \mid y_{1:t-1})$.

However, the likelihood estimator (9) is valid for the general auxiliary particle filter, given in Algorithm 2, as well. Derive this estimator for the auxiliary particle filter, in a similar fashion as was done above for the bootstrap particle filter.

Hint: You need to take the auxiliary variables into account. That is, write the pdf $p(y_t \mid y_{1:t-1})$ *as an integral over* (x_t, a_t) *(more precisely, an integral over* x_t *and sum over* a_t*). Then interpret this integral as an expected value with respect to the joint proposal used in the auxiliary particle filter.*

N.B The expression (9) *assumes that the weights are computed as stated in Algorithm 2, i.e. that the* unnormalized *weights at time t,* $\widetilde{w}_t$*, are expressed in terms of the* normalized *weights at time* $t-1$, w_{t-1}.

Algorithm 1 Bootstrap particle filter (for $i = 1, \ldots, N$)

(a) **Initialization ($t = 0$):**
   i. Sample $x_0^i \sim p(x_0)$.
   ii. Set initial weights: $w_0^i = 1/N$.

(b) **for** $t = 1$ **to** T **do**
   i. **Resample:** sample ancestor indices $a_t^i \sim \mathcal{C}(\{w_{t-1}^j\}_{j=1}^N)$.
   ii. **Propagate:** sample $x_t^i \sim p(x_t \mid x_{t-1}^{a_t^i})$.
   iii. **Weight:** compute $\widetilde{w}_t^i = p(y_t \mid x_t^i)$ and normalize $w_t^i = \widetilde{w}_t^i / \sum_{j=1}^N \widetilde{w}_t^j$.

Algorithm 2 Auxiliary particle filter (for $i = 1, \ldots, N$)

(a) **Initialization ($t = 0$):**
   i. Sample $x_0^i \sim p(x_0)$.
   ii. Set initial weights: $w_0^i = 1/N$.

(b) **for** $t = 1$ **to** T **do**
   i. **Resample:** sample ancestor indices $a_t^i \sim \mathcal{C}(\{\nu_{t-1}^j\}_{j=1}^N)$.
   ii. **Propagate:** sample $x_t^i \sim q(x_t \mid x_{t-1}^{a_t^i}, y_t)$.
   iii. **Weight:** compute

$$\widetilde{w}_t^i = \frac{w_{t-1}^{a_t^i}}{\nu_{t-1}^{a_t^i}} \frac{p(y_t \mid x_t^i) p(x_t^i \mid x_{t-1}^{a_t^i})}{q(x_t^i \mid x_{t-1}^{a_t^i}, y_t)}$$

   and normalize $w_t^i = \widetilde{w}_t^i / \sum_{j=1}^N \widetilde{w}_t^j$.

II.4 Forgetting

Consider the bootstrap particle filter for the LGSS model

$$X_t = 0.7 X_{t-1} + V_t, \qquad V_t \sim \mathcal{N}(0, Q), \tag{11a}$$

$$Y_t = 0.5 X_t + E_t, \qquad E_t \sim \mathcal{N}(0, 1). \tag{11b}$$

Let the initial state be distributed according to $X_0 \sim \mathcal{N}(0, 1)$.

but modify the model to $Q = 0$ instead. What happens to the errors in the particle filter (compared to the Kalman filter, the exact solution) along the time dimension? Specifically, run the particle filter, say, 100 times (using a fixed N) for the same data and compute the mean-squared-error of the test function $\varphi(x_t) = x_t$ with respect to the Kalman filter solution,

$$\frac{1}{100} \sum_{\ell=1}^{100} \left(\widehat{I}_{t,N}^{\text{PF},\ell}(\varphi) - \mathbb{E}[X_t \mid y_{1:t}] \right)^2$$

for each time step $t = 1, 2, \ldots$, where $I_{t,N}^{\text{PF},\ell}(\varphi)$ is the estimate of $\mathbb{E}[X_t \mid y_{1:t}]$ obtained from the ℓth run of the particle filter. Is the particle filter stable?