Particle Filtering for the Log–Normal SV model

Leopoldo Catania
Aarhus University and CREATES

leopoldo.catania@econ.au.dk

Filtering in State Space models

1) Filtering with linear and Gaussian State Space models is routinely done via a run of the Kalman filter.

2) Filtering with linear non–Gaussian State Space models can be still done with the Kalman filter, however filtered estimates are not optimal.

3) Filtering with non–linear (possibly) non–Gaussian state space models does not have a standard solution. Possibilities are:
   - 3a) Linearize the system and apply 2) as we have done with the SV model.
   - 3b) In some cases a modified Kalman filter can be developed, for instance: i) The Unscented Kalman Filter, and ii) the Extended Kalman Filter.
   - 3c) Develop a simulation based filtering procedure.

Today we focus on 3c) and apply the Bootstrap particle filter of Gordon et al. (1993) to the Log–Normal Stochastic Volatility model.

Law of Large Numbers

Let $Y_1, Y_2, \ldots, Y_n$ be a sequence of n iid random variables with finite mean $E[Y_1] = E[Y_2] = \cdots = E[Y_n] = \mu$. Let

$$\bar{Y}_n = \frac{1}{n}(Y_1 + Y_2 + \cdots + Y_n),$$

be the sample average of the sequence of random variables.
There are two versions of the Law of Large Numbers (LLN):

- The weak LLN implies that:

$$\lim_{n\to\infty} \bar{Y}_n \xrightarrow{p} \mu,$$

  i.e. $\lim_{n\to\infty} P(|\bar{Y}_n - \mu| > \varepsilon) = 0$ for any $\varepsilon > 0$.

- The strong LLN implies that:

$$\lim_{n\to\infty} \bar{Y}_n \xrightarrow{a.s.} \mu,$$

  i.e. $P(\lim_{n\to\infty} \bar{Y}_n = \mu) = 1$.

Importance Sampling

Suppose that the random variable Y is distributed with density $p(y)$. Assume we are interested in calculating some moment of Y, for instance $E[x(Y)]$ which we are not able to compute. Of course it is true that:

$$E[x(Y)] = \int_{y \in \mathcal{Y}} x(y)p(y)dy,$$

Evidently, if the dimension of Y is large (say 5 or 6) this integral cannot be computed numerically. By the law of large number we could for example approximate the integral by sampling random draws $y^{(i)}$, for $i = 1, \ldots, N$, from the distribution of Y and calculate:

$$E[x(Y)] = \frac{1}{N} \sum_{i=1}^{N} x(y^{(i)}),$$

However, if sampling $y^{(i)}$ is not possible (or too costly from a computational point of view) Monte Carlo approximation cannot be done.

Importance Sampling

Consider now another density $g(y)$ which can be evaluated and is chosen to resemble $p(y)$ as closely as is reasonably possible while being easy to sample from. We can write:

$$\begin{aligned} E[x(Y)] &= \int_{y \in \mathcal{Y}} x(y) p(y) dy, \\ &= \int_{y \in \mathcal{Y}} x(y) \frac{p(y)}{g(y)} g(y) dy \\ &= E_g \left[x(y) \frac{p(y)}{g(y)} \right], \end{aligned}$$

where $E_g[\cdot]$ means that expectation is taken with respect to the **importance** density $g(y)$.

Importance Sampling

Since:

$$E[x(Y)] = E_g\left[x(y)\frac{p(y)}{g(y)}\right],$$

we can now approximate by Monte Carlo simulation $E_g\left[x(y)\frac{p(y)}{g(y)}\right]$, that is:

$$E_g\left[x(y)\frac{p(y)}{g(y)}\right] = \frac{1}{N}\sum_{i=1}^{N} x(y^{(i)})\frac{p(y^{(i)})}{g(y^{(i)})},$$

where observations $y^{(i)}$ have been sampled according to the density $g(y)$.

Filtering in State Space models

Filtering in state space models means recovering the (unobserved) signal from the data. For instance in a SV model it means recovering the latent volatility from the data.

Consider the non–linear possibly non–Gaussian state space model:

$$\begin{aligned} p(y_t|Z(\alpha_t), \varepsilon_t), \quad & \varepsilon_t \sim p(\varepsilon_t) \\ \alpha_{t+1} = T(\alpha_t) + R(\alpha_t)\eta_t, \quad & \eta_t \sim p(\eta_t), \end{aligned}$$

for $t = 1, 2, \ldots,$ where Z, T, and R are known functions of α_t and where ε_t and η_t are disturbance series. Here y_t represents the observed variable and α_t the (unobserved) state variable.

Filtering coincides with the evaluation of:

$$\bar{x}_t = E[x_t(\alpha_{1:t})|y_{1:t}], \qquad \text{for all } t \text{ and some function } x_t(\cdot).$$

Filtering in State Space models

Having a closer look at the filtered estimate $\hat{\alpha}_{1:t}$ reveals the main problem we face with nonlinear State Space models:

$$E[x_t(\alpha_{1:t})|y_{1:t}] = \int x_t(\alpha_{1:t}) p(\alpha_{1:t}|y_{1:t}) d\alpha_{1:t},$$

which is numerically unfeasible to solve.

Filtering by direct Importance Sampling

We can try to apply Importance Sampling in order to solve $E[x_t(\alpha_{1:t})|y_{1:t}]$. Consider an importance density $g(\alpha_{1:t}|y_{1:t})$, we can now write:

$$\begin{aligned} E[x_t(\alpha_{1:t})|y_{1:t}] &= \int x_t(\alpha_{1:t}) \frac{p(\alpha_{1:t}|y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} g(\alpha_{1:t}|y_{1:t}) d\alpha_{1:t} \\ &= E_g \left[x_t(\alpha_{1:t}) \frac{p(\alpha_{1:t}|y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} |y_{1:t} \right], \end{aligned}$$

where E_g denotes expectation with respect to the importance density $g(\alpha_{1:t}|y_{1:t})$.

Filtering by direct Importance Sampling

Since $p(\alpha_{1:t}, y_{1:t}) = p(\alpha_{1:t}|y_{1:t})p(y_{1:t})$ we can write:

$$\bar{x}_t = \frac{1}{p(y_{1:t})} E_g \left[x_t(\alpha_{1:t}) \tilde{\omega}_t | y_{1:t} \right],$$

where

$$\tilde{\omega}_t = \frac{p(\alpha_{1:t}, y_{1:t})}{g(\alpha_{1:t}|y_{1:t})}.$$

Note that generally we don't know how to evaluate $p(y_{1:t})$! However, setting $x_t(\alpha_{1:t}) = 1$ we see that:

$$1 = \frac{1}{p(y_{1:t})} E_g \left[\tilde{\omega}_t | y_{1:t} \right] \quad \Rightarrow \quad p(y_{1:t}) = E_g \left[\tilde{\omega}_t | y_{1:t} \right],$$

thus:

$$\bar{x}_t = \frac{E_g \left[x_t(\alpha_{1:t}) \tilde{\omega}_t | y_{1:t} \right]}{E_g \left[\tilde{\omega}_t | y_{1:t} \right]},$$

Filtering by direct Importance Sampling

Having a sample of draws $\alpha_{1:t}^{(1)}, \ldots, \alpha_{1:t}^{(N)}$ from the importance density $g(\alpha_{1:t}|y_{1:t})$, we can approximate $\bar{x}_t$ by:

$$\hat{x}_t = \frac{N^{-1} \sum_{i=1}^{N} x_t(\alpha_{1:t}^{(i)}) \tilde{\omega}_t^{(i)}}{N^{-1} \sum_{i=1}^{N} \tilde{\omega}_t^{(i)}},$$

that is:

$$\hat{x}_t = \sum_{i=1}^{N} x_t(\alpha_{1:t}^{(i)}) \omega_t^{(i)},$$

where:

$$\tilde{\omega}_t^{(i)} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{g(\alpha_{1:t}^{(i)}|y_{1:t})}, \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

The values $\tilde{\omega}_t^{(i)}$ are called "importance weights" and the values $\omega_t^{(i)}$ are called "normalised importance weights". Note that $p(\alpha_{1:t}^{(i)}, y_{1:t}) = p(y_{1:t}|\alpha_{1:t}^{(i)}) p(\alpha_{1:t}^{(i)})$ which is easy to compute.

Filtering by direct Importance Sampling

Consider the SV model:

$$y_t = \exp(\alpha_t/2)\varepsilon_t$$
$$\alpha_t = \omega + \phi\alpha_{t-1} + \tau\eta_t,$$

where ε_t and η_t are uncorrelated iid normally distributed shocks.

Suppose we have consistently estimated ω, ϕ, and τ by the GMM or QML estimators using a sample of T observations.

We now want to recover $\hat{\sigma}_t = E[\exp(\alpha_t/2)|y_{1:t}]$ for all $t = 1, \ldots, T$.

Filtering by direct Importance Sampling

To apply the IS we specify $x_t(\alpha_{1:t}) = \exp(\alpha_t/2)$. For the importance density $g(\alpha_{1:t}|y_{1:t})$ we choose $p(\alpha_{1:t})$, where $p(\alpha_{1:t})$ is computed according to the SV model. We thus have:

$$\tilde{\omega}_t^{(i)} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{g(\alpha_{1:t}^{(i)}|y_{1:t})} = \frac{p(\alpha_{1:t}^{(i)}, y_{1:t})}{p(\alpha_{1:t}^{(i)})} = \frac{p(y_{1:t}|\alpha_{1:t}^{(i)})p(\alpha_{1:t}^{(i)})}{p(\alpha_{1:t}^{(i)})} = p(y_{1:t}|\alpha_{1:t}^{(i)}),$$

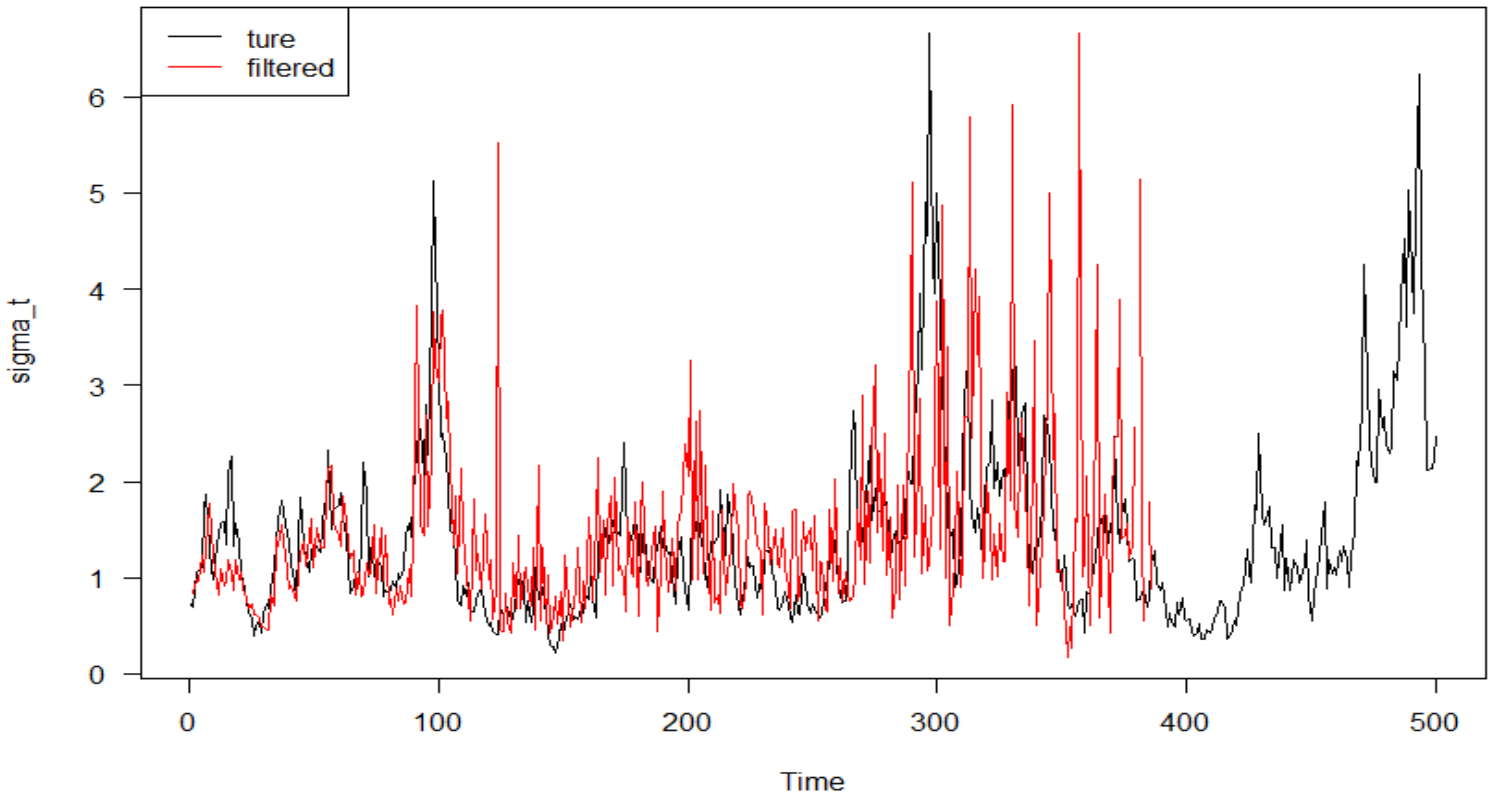
and

$$\omega_t^{(i)} = \frac{p(y_{1:t}|\alpha_{1:t}^{(i)})}{\sum_{j=1}^{N} p(y_{1:t}|\alpha_{1:t}^{(j)})}$$

Note that:

1) This is very demanding from a computational point of view.
2) Our importance density does not include the information contained in the data: $g(\alpha_{1:t}|y_{1:t}) = p(\alpha_{1:t})$. This means that the chosen importance density can be "far" from the target density thus reducing the efficiency of the sampler. Lower efficiency implies a higher number of draws N.

Filtering by direct Importance Sampling



Filtering of $\sigma_t = \exp(\alpha_t/2)$ with direct importance sampling using $g(\alpha_{1:t}|y_{1:t}) = p(\alpha_{1:t})$ and $N = 5000$. The algorithm broke before observation at time $t = 400$. Note how the filtering procedure becomes less accurate as long as t increases. The computational time is about 25 minutes.

Sequential Importance Sampling

To improve the simple method of filtering, it seems more natural in the context of filtering to retain the previous selection of $\alpha_{1:t-1}^{(i)}$ for each i and to confine the new sampling at time t to the selection of $\alpha_t^{(i)}$ only.

We call this sequential process for choosing $\alpha_{1:t}^{(i)}$ and the estimation based on it "particle filtering"; the resulting sets of values $\alpha_{1:t}^{(1)}, \ldots, \alpha_{1:t}^{(N)}$ are called "particles".

Thus, the ith particle at time t is defined by the relation:

$$\alpha_{1:t}^{(i)} = (\alpha_{1:t-1}^{(i)}, \alpha_t^{(i)}),$$

where $\alpha_{1:t-1}^{(i)}$ is the ith particle at time $t-1$.

Sequential Importance Sampling

Consider the importance density $g(\alpha_{1:t}|y_{1:t})$. We can write:

$$\begin{aligned} g(\alpha_{1:t}|y_{1:t}) &= \frac{g(\alpha_{1:t}, y_{1:t})}{g(y_{1:t})} \\ &= \frac{g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}, y_{1:t})}{g(y_{1:t})} \\ &= g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}|y_{1:t}). \end{aligned}$$

Now suppose that $\alpha_{1:t-1}$ is selected using knowledge only of $y_{1:t-1}$. Moreover, given the realised values of $\alpha_{1:t-1}$ and $y_{1:t-1}$, the value of the observable y_t has already been selected by a process which does not depend on the simulated sequence $\alpha_{1:t-1}$. Under these circumstances, the density $g(\alpha_{1:t-1}|y_{1:t})$ is not affected by including y_t in its set of conditional variables $y_{1:t}$. Hence, $g(\alpha_{1:t-1}|y_{1:t}) \equiv g(\alpha_{1:t-1}|y_{1:t-1})$.

Sequential Importance Sampling

If $g(\alpha_{1:t-1}|y_{1:t}) \equiv g(\alpha_{1:t-1}|y_{1:t-1})$, then:

$$g(\alpha_{1:t}|y_{1:t}) = g(\alpha_t|\alpha_{1:t-1}, y_{1:t})g(\alpha_{1:t-1}|y_{1:t-1}),$$

such that we can specify an updating recursion from $g(\alpha_{1:t-1}|y_{1:t-1})$ to $g(\alpha_{1:t}|y_{1:t})$.

Note that it is assumed that sampling $\alpha_t^{(i)}$ from $g(\alpha_t|\alpha_{1:t-1}, y_{1:t})$ is practical and inexpensive.

Sequential Importance Sampling

We now develop the recursion for the importance weights $\tilde{\omega}_t^{(i)}$ which are required to approximate $E[x_t(\alpha_{1:t})|y_{1:t}]$.

We have that:

$$\begin{aligned}\tilde{\omega}_t &= \frac{p(\alpha_{1:t}, y_{1:t})}{g(\alpha_{1:t}|y_{1:t})} \\ &= \frac{p(\alpha_{1:t-1}, y_{1:t-1})p(\alpha_t, y_t|\alpha_{1:t-1}, y_{1:t-1})}{g(\alpha_{1:t-1}|y_{1:t-1})g(\alpha_t|\alpha_{1:t-1}, y_{1:t})},\end{aligned}$$

Due to the Markovian nature of the model we are considering we have:

$$p(\alpha_t, y_t|\alpha_{1:t-1}, y_{1:t-1}) = p(\alpha_t|\alpha_{t-1})p(y_t|\alpha_t),$$

such that for the ith particle we have:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad \text{and} \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

Sequential Importance Sampling

The recursion is initialized by $\tilde{\omega}_0^{(i)} = 1$ for $i = 1, \ldots, N$. In most practical situation we have that $x_t(\alpha_{1:t}) = x(\alpha_t)$, for example $x(\alpha_t) = \exp(\alpha_t/2)$ in the case of volatility filtering. Let's continue with this convention. To filter $\exp(\alpha_t/2)$ we apply the Monte Carlo approximation:

$$\widehat{\exp(\alpha_t/2)} = \sum_{i=1}^{N} \exp(\alpha_t^{(i)}/2)\omega_t^{(i)},$$

where $\alpha_t^{(i)}$ is selected from the importance density $g(\alpha_t|\alpha_{1:t-1}, y_{1:t})$.

Importance sampling based on this approach is called "sequential importance sampling" (SIS). It was originally developed by Hammersley and Morton (1954) and applied to state space models by Handschin and Mayne (1969) and Handschin (1970).

Sequential Importance Sampling: Algorithm

(i) Sample α_t: draw N values $\alpha_t^{(i)}$ from $g(\alpha_t|\alpha_{t-1}^{(i)}, y_{1:t})$ and store $\alpha_{t-1:t}^{(i)} = \{\alpha_{t-1}^{(i)}, \alpha_t^{(i)}\}$.

(ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad i = 1, \ldots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$.

(iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^{N} x_t(\alpha_t^{(i)})\omega_t^{(i)}.$$

Degeneracy

As t increases, the distribution of the weights $\omega_t^{(i)}$ based on the recursion:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y_{1:t})}, \quad \text{and} \quad \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}},$$

becomes highly skewed. It is possible for all but one particle to have negligible weights, for t large. We then say that the sample has become "degenerate".

The problem of degeneracy typically occurs when the likelihood function $p(y_t|\alpha_t)$ is highly peaked relatively to the density $p(\alpha_t|\alpha_{t-1})$, with the effect that few of the values $\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}$ lead to non–neglible values of $\omega_t^{(i)}$. It is obviously wasteful to retain particles in the recursion that are contributing negligible weights to the estimate $E[x(\alpha_t)|y_{1:t}]$.

Resampling

A way to combat degeneracy is to resample N particles $\tilde{\alpha}_t^{(i)}$, $i = 1, \ldots, N$ with replacement from the set of particles $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$ according to probabilities $\omega_t^{(1)}, \ldots, \omega_t^{(N)}$ at the end of each iteration of the Sequential Importance Sampling.

Although resampling is a technique that combats degeneracy in particle filtering, it clearly introduces additional Monte Carlo variation into the estimate of $E[x(\alpha_t)|y_{1:t}]$. A solution to mitigate the increased variability is to resample only at some iterations of the algorithm, see Liu and Chen (1998) for further details.

Sequential Importance Sampling algorithms that incorporates a resampling step are called "Sequential Importance Sampling Resampling" (SISR).

After a step of resampling the importance weights are set to $\tilde{\omega}_t^{(i)} = 1$ for all i.

Sequential Importance Sampling Resampling: Algorithm

(i) Sample α_t: draw N values $\alpha_t^{(i)}$ from $g(\alpha_t|\tilde{\alpha}_{t-1}^{(i)}, y_{1:t})$ and store $\alpha_{t-1:t}^{(i)} = \{\tilde{\alpha}_{t-1}^{(i)}, \alpha_t^{(i)}\}$.

(ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} \frac{p(\alpha_t^{(i)}|\tilde{\alpha}_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\tilde{\alpha}_{1:t-1}^{(i)}, y_{1:t})}, \quad i = 1, \ldots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$. (note that $\tilde{\omega}_{t-1}^{(i)} = 1$)

(iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^{N} x_t(\alpha_t^{(i)})\omega_t^{(i)}.$$

(iv) Resample: draw N new independent particles $\tilde{\alpha}_t^{(i)}$ from $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$ with replacement and with corresponding probabilities $\{\omega_t^{(1)}, \ldots, \omega_t^{(N)}\}$. Set $\tilde{\omega}_t^{(i)} = 1$ for all i.

The bootstrap particle filter

The first particle filter to be developed was the bootstrap filter of Gordon et al. (1993). It is sometimes called "Sampling Importance Resampling" algorithm.

The bootstrap filter uses the proposal distribution $g(\alpha_t^{(i)}|\tilde{\alpha}_{1:t-1}^{(i)}, y_{1:t}) = p(\alpha_t^{(i)}|\tilde{\alpha}_{t-1}^{(i)})$. This implies that:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} p(y_t|\alpha_t^{(i)}),$$

in the step (ii) of the Sequential Importance Sampling Resampling algorithm.

At first sight this looks crude since it neglects relevant information in y_t but when used with resampling and with N large enough, it can work well in many cases of interest and it is widely used in practice.

The bootstrap particle filter: Algorithm

(i) Sample α_t: draw N values $\alpha_t^{(i)}$ from $p(\alpha_t|\tilde{\alpha}_{t-1}^{(i)})$ and store $\alpha_{t-1:t}^{(i)} = \{\tilde{\alpha}_{t-1}^{(i)}, \alpha_t^{(i)}\}$.

(ii) Weights: compute the corresponding weights $\tilde{\omega}_t^{(i)}$:

$$\tilde{\omega}_t^{(i)} = \tilde{\omega}_{t-1}^{(i)} p(y_t|\alpha_t^{(i)}), \quad i = 1, \ldots, N,$$

and normalize the weights to obtain $\omega_t^{(i)}$. (note that $\tilde{\omega}_{t-1}^{(i)} = 1$)

(iii) Compute the variable of interest $x_t(\alpha_t)$: given the set of particles $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$, compute

$$\widehat{x_t(\alpha_t)} = \sum_{i=1}^{N} x_t(\alpha_t^{(i)}) \omega_t^{(i)}.$$

(iv) Resample: draw N new independent particles $\tilde{\alpha}_t^{(i)}$ from $\{\alpha_t^{(1)}, \ldots, \alpha_t^{(N)}\}$ with replacement and with corresponding probabilities $\{\omega_t^{(1)}, \ldots, \omega_t^{(N)}\}$. Set $\tilde{\omega}_t^{(i)} = 1$ for all i.

The bootstrap particle filter: Algorithm

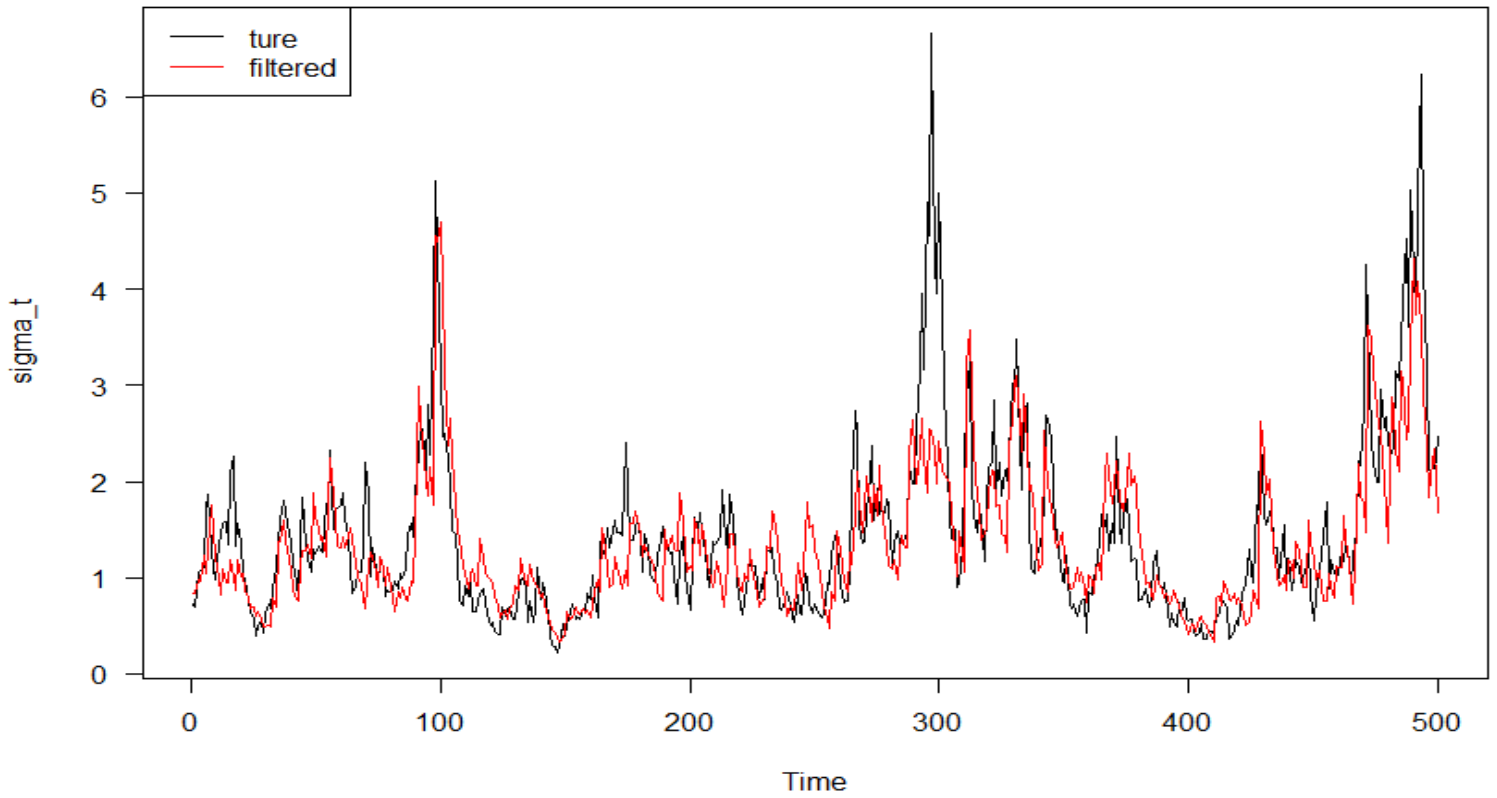
Note that, since we are resampling at each iteration step (ii) of the Bootstrap algorithm coincides with $\tilde{\omega}_t^{(i)} = p(y_t|\alpha_t^{(i)})$ (because $\tilde{\omega}_{t-1}^{(i)} = 1$).

Since resampling at each iteration increases the Monte Carlo variation in the computation of $E[x_t(\alpha_{1:t}|y_{1:t})]$, Carpenter et al. (1999) and the Liu and Chen (1998) have proposed different resempling schemes. The most popular one is based on the evaluation of the *effective sample size*:

$$ESS = \left(\sum_{i=1}^{N} {\omega_t^{(i)}}^2 \right)^{-1}.$$

Then resampling occurs only when $ESS < gN$ with $g = 0.75$ or $g = 0.5$.

Filtering with the bootstrap particle filter



Filtering of $\sigma_t = \exp(\alpha_t/2)$ with the bootstrap particle filter and $N = 10000$. The computational time is about 1.6 seconds.

Extensions

There are many ways to improve the filtering procedure of a non–linear non–Gaussian state space models. Most of these extensions aim at finding a better importance distribution $g(\alpha_t|\alpha_{1:t-1}, y_{1:t})$. Further readings can be find in Durbin and Koopman (2012), Creal (2012), Andrieu et al. (2001), and Doucet et al. (2001).

References I

Andrieu, C., Doucet, A., and Punskaya, E. (2001). Sequential monte carlo methods for optimal filtering. In *Sequential Monte Carlo Methods in Practice*, pages 79–95. Springer.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

Creal, D. (2012). A survey of sequential monte carlo methods for economics and finance. *Econometric reviews*, 31(3):245–296.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*, volume 38. Oxford University Press.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET.

References II

Hammersley, J. M. and Morton, K. W. (1954). Poor man's monte carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 23–38.

Handschin, J. (1970). Monte carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6(4):555–563.

Handschin, J. E. and Mayne, D. Q. (1969). Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International journal of control*, 9(5):547–559.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.