Sequential Monte Carlo (SMC) Methods (Pure Filter)

Hedibert Freitas Lopes

The University of Chicago Booth School of Business
5807 South Woodlawn Avenue, Chicago, IL 60637
http://faculty.chicagobooth.edu/hedibert.lopes

hlopes@ChicagoBooth.edu

Outline

Nonnormal & nonlinear dynamic models

Most nonnormal and nonlinear dynamic models are defined by

- Observation equation

$$p(y_{t+1}|x_{t+1},\theta)$$

- System or evolution equation

$$p(x_{t+1}|x_t,\theta)$$

- Initial distribution

$$p(x_0|\theta)$$

The fixed parameters that drive the state space model, θ, is kept known and omitted for now.

Forward filtering

Posterior at time t:

$$p(x_t|y^t).$$

Prior at time $t+1$:

$$\underbrace{p(x_{t+1}|y^t)}_{\text{prior at t}} = \int \underbrace{p(x_{t+1}|x_t)}_{\text{evolution}} \underbrace{p(x_t|y^t)}_{\text{posterior at t-1}} dx_t$$

Posterior at time $t+1$:

$$p(x_{t+1}|y^{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y^t)$$

These densities are usually unavailable in closed form.

Boostrap filter (BF)

Gordon, Salmond and Smith's (1993) seminal paper uses SIR to obtain draws from $p(x_{t+1}|y^{t+1})$ based on draws from $p(x_t|y^t)$.

Let $x_t^{(i)}$ be a draw from $p(x_t|y^t)$, for $i = 1, \ldots, N$.
Let $\tilde{x}_{t+1}^{(i)}$ be a draw from $p(x_{t+1}|x_t^{(i)})$, for $i = 1, \ldots, N$.
Then $\tilde{x}_{t+1}^{(i)}$ is a draw from $p(x_{t+1}|y^{t+1})$, for $i = 1, \ldots, N$.

SIR argument: Sample k^i from $\{1, \ldots, M\}$ with (unnormalized) weights

$$\omega_{t+1}^{(j)} \propto p(y_{t+1}|\tilde{x}_{t+1}^{(j)})$$

and let $x_{t+1}^{(i)} = \tilde{x}_{t+1}^{(k^i)}$.

Then $x_{t+1}^{(i)}$ is a draw from $p(x_{t+1}|y^{t+1})$, for $i = 1, \ldots, N$.

SIS with Resampling (SISR)

$$\{x_t^1, \ldots, x_t^N\} \sim p(x_t|y^t)$$

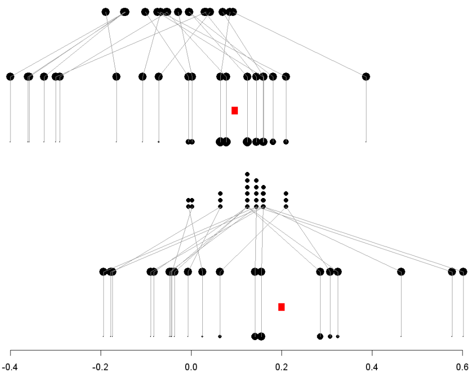
$$\tilde{x}_{t+1}^i \sim p(x_{t+1}|x_t^i)$$

$$\omega_{t+1}^i \propto p(y_{t+1}|\tilde{x}_{t+1}^i)$$

$$\{x_{t+1}^1, \ldots, x_{t+1}^N\} \sim p(x_{t+1}|y^{t+1})$$

$$\tilde{x}_{t+2}^i \sim p(x_{t+2}|x_{t+1}^i)$$

$$\omega_{t+2}^i \propto p(y_{t+2}|\tilde{x}_{t+2}^i)$$



Uniform weights is the goal!

Resampling or not?

Theoretically, the resampling step is not necessary. Within a given time t, resampling always increases the variability of estimators.

For instance, let

$$I_1 = \sum_{i=1}^{N} h(\tilde{x}_t^{(i)}) w_t^{(i)} \quad \text{and} \quad I_2 = \frac{1}{N} \sum_{i=1}^{N} h(x_t^{(i)})$$

be two MC estimators of $E(h(x_t)|y^t)$ with I_1 based on (normalized) weights

$$w_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{j=1}^{N} \omega_t^{(j)}}.$$

It can be shown (Raoblackwellization) that

$$V(I_1) \leq V(I_2).$$

Effective sample size

Liu and Chen (1995, 1998) argue that resampling at every time t is usually neither necessary nor efficient since it induces excessive variations.

Kong *et al.* (1994) and Liu (1996) proposed resampling whenever the effective sample size

$$N_{\mathrm{eff},t} = \frac{1}{\sum_{i=1}^{N} \left(w_t^{(i)} \right)^2}$$

is less than a certain threshold.

Example 1: Local level model

The model is

$$
\begin{aligned}
y_t|x_t &\sim N(x_t, \sigma^2) \\
x_t|x_{t-1} &\sim N(x_{t-1}, \tau^2)
\end{aligned}
$$

with $(x_0|y^0) \sim N(m_0, C_0)$.

If $(x_{t-1}|y^{t-1}) \sim N(m_{t-1}, C_{t-1})$, then

$$(x_t|y^{t-1}) \sim N(m_{t-1}, R_t)$$

where $R_t = C_{t-1} + \tau^2$ and

$$(x_t|y^t) \sim N(m_t, C_t)$$

where $m_t = (1 - A_t)m_{t-1} + A_t y_t$, $C_t = A_t\sigma^2$ and $A_t = R_t/(R_t + \sigma^2)$.

Example 1: SIS and bootstrap filters

Sequential importance sampling (SIS):

- $\{(x_{t-1}, \omega_{t-1})^{(i)}\}_{i=1}^{N} \sim p(x_{t-1}|y^{t-1})$.
- $\{(\tilde{x}_t, \omega_{t-1})^{(i)}\}_{i=1}^{N} \sim p(x_t|y^{t-1})$, where

$$\tilde{x}_t^{(i)} \sim N(x_{t-1}^{(i)}, \tau^2).$$

- $\{(\tilde{x}_t, \omega_t)^{(i)}\}_{i=1}^{N} \sim p(x^t|y^t)$, where

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} f_N(y_t; \tilde{x}_t^{(i)}, \sigma^2).$$

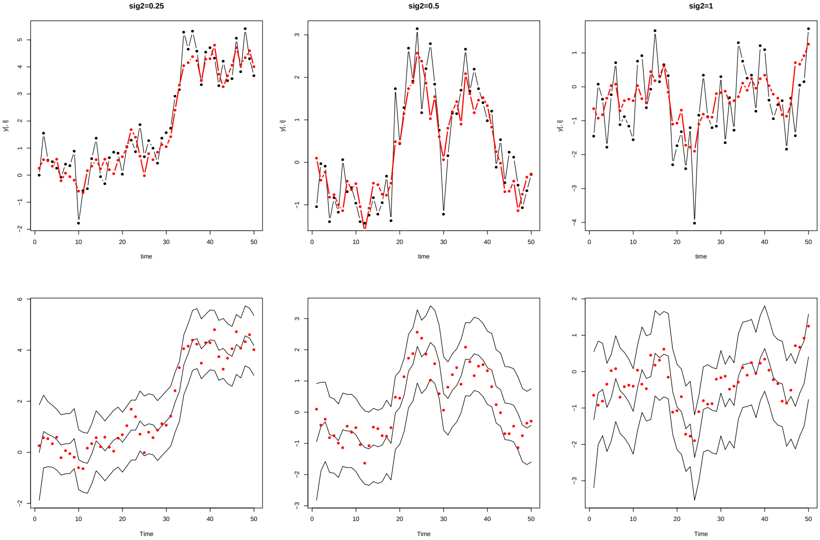
Resampling:

Resample $\{x_t^{(1)}, \ldots, x_t^{(N)}\}$ from $\{\tilde{x}_t^{(1)}, \ldots, \tilde{x}_t^{(N)}\}$ with (normalized) weights $\{w_t^{(1)}, \ldots, w_t^{(N)}\}$.

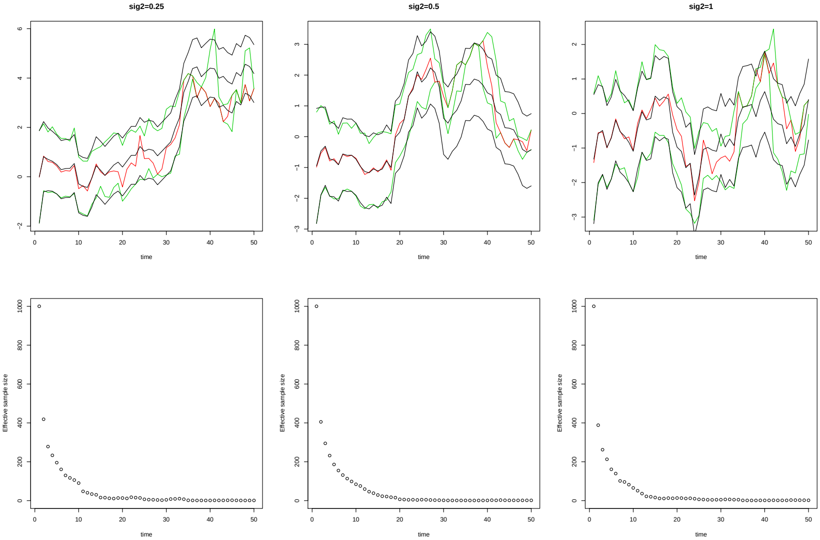
In this case, $\{(x_t, \omega_t)^{(i)}\}_{i=1}^{N} \sim p(x_t|y^t)$ with weights $\omega_t \propto 1$.

Example 1: local level model

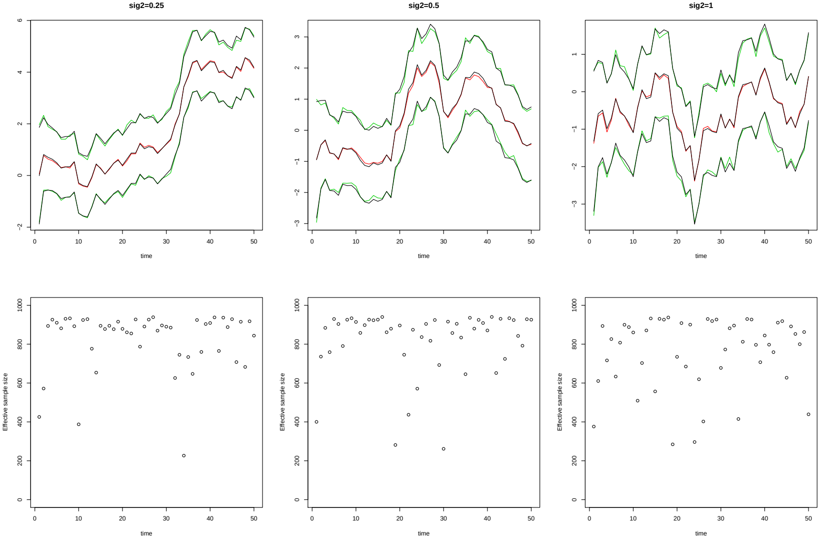
$n = 50,\ x_0 = 0,\ \tau^2 = 0.5$ and $\sigma^2 = (0.25, 0.5, 1.0)$.



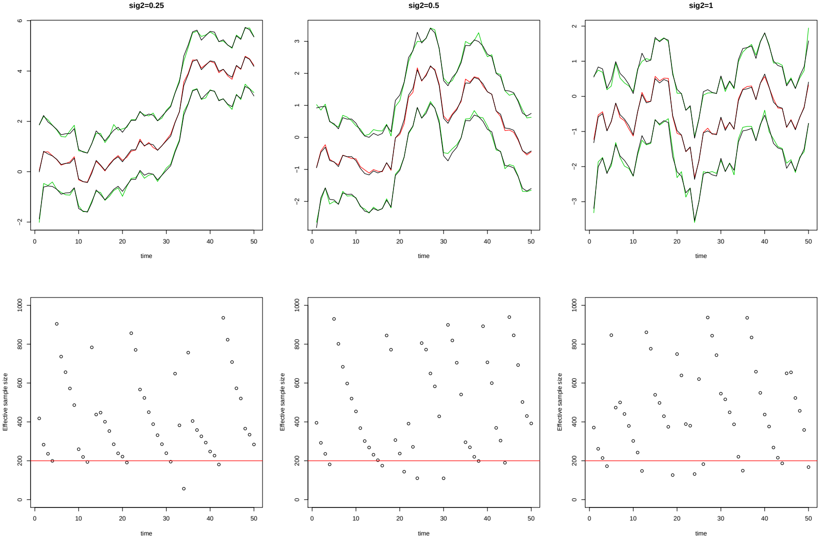
SIS filter



Bootstrap filter

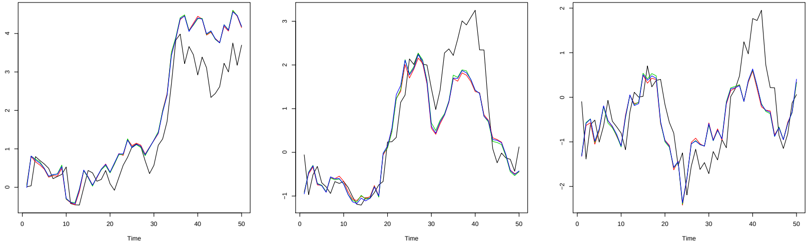


$\text{SIS}_{0.2}$: SIS filter with resampling when $N_{eff} < 0.2N$



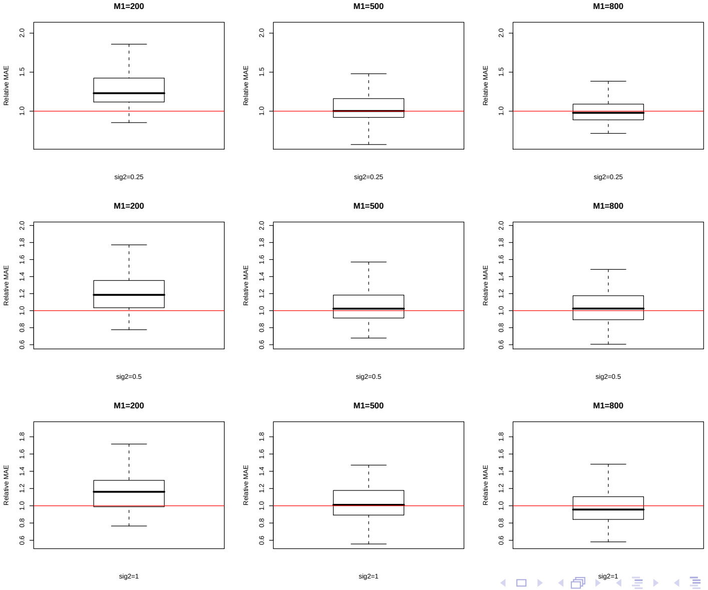
SIS, BF, $SIS_{0.2}$

Comparing estimates of $E(x_t|y^t)$.

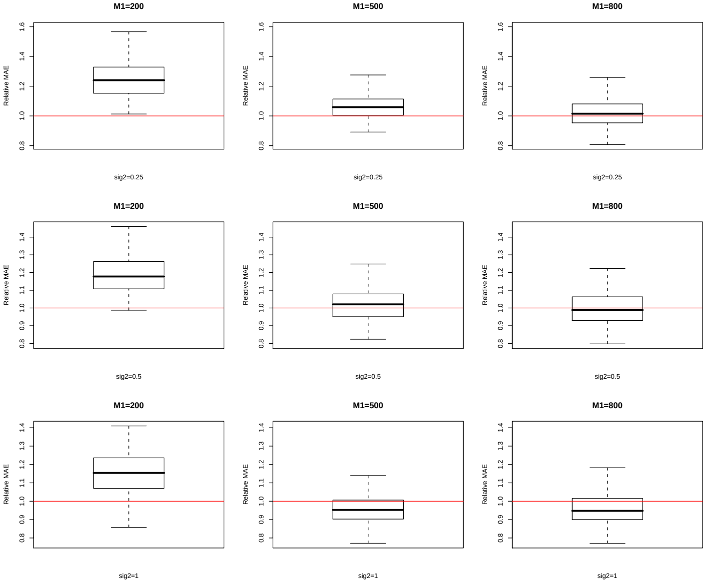


Comparing BF and $\mathrm{SIS}_{0.2}$ when $n = 50$

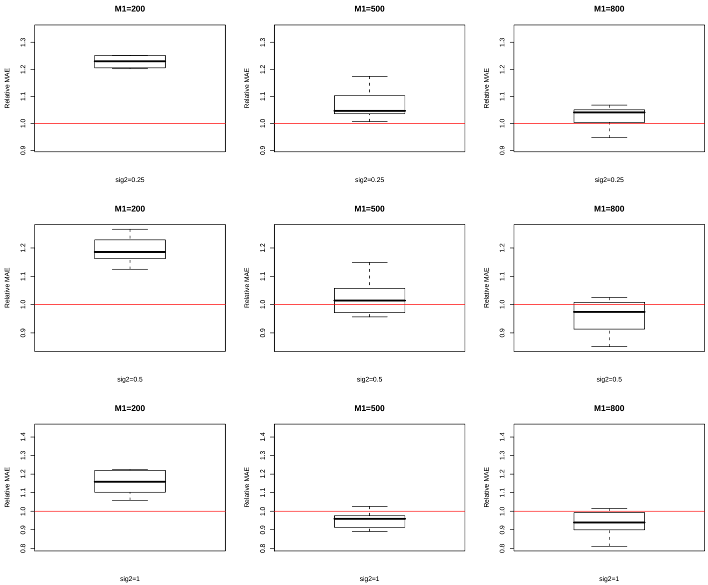
$$\mathrm{MAE}=\sum_{i=1}^{n} |x_t - \widehat{E}(x_t|y^t)|/n; \ \mathrm{RMAE} = \mathrm{MAE}_{bf}/\mathrm{MAE}_{sis}$$



Comparing BF and $SIS_{0.2}$ when $n = 200$



Comparing BF and $SIS_{0.2}$ when $n = 500$



Auxiliary particle filter (APF)

Recall the two main steps in any dynamic model:

$$\begin{aligned} p(x_t|y^{t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})dx_{t-1} \\ p(x_t|y^t) &\propto p(y_t|x_t)p(x_t|y^{t-1}) \\ &= \int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})dx_{t-1} \end{aligned}$$

Based on $\left\{(x_{t-1}, \omega_{t-1})^{(i)}\right\}_{i=1}^{N} \sim p(x_{t-1}|y^{t-1})$:

$$\hat{p}(x_t|y^{t-1}) \propto \sum_{i=1}^{N} p(x_t|x_{t-1}^{(i)})\omega_{t-1}^{(i)}$$

and

$$\hat{p}(x_t|y^t) \propto \sum_{i=1}^{N} p(y_t|x_t)p(x_t|x_{t-1}^{(i)})\omega_{t-1}^{(i)}.$$

Pitt and Shephard's (1999) idea

The previous mixture approximation suggests an augmentation scheme where the new target distribution is

$$\hat{p}(x_t, k|y^t) \propto p(y_t|x_t)p(x_t|x_{t-1}^{(k)})\omega_{t-1}^{(k)}.$$

A natural proposal distribution is

$$q(x_t, k|y^t) \propto p(y_t|g(x_{t-1}^{(k)}))p(x_t|x_{t-1}^{(k)})\omega_{t-1}^{(k)}$$

where, for instance, $g(x_{t-1}) = E(x_t|x_{t-1})$.

By a simple SIR argument, the weight of the particle x_t is

$$\omega_t \propto \frac{p(y_t|x_t)}{p(y_t|g(x_{t-1}))}$$

APF algorithm

- $\left\{(x_{t-1}, \omega_{t-1})^{(i)}\right\}_{i=1}^{N}$ summarizes $p(x_{t-1}|y^{t-1})$.
- For $j = 1, \ldots, N$
  - Draw k^j from $\{1, \ldots, N\}$ with weights $\{\tilde{\omega}_{t-1}^{(1)}, \ldots, \tilde{\omega}_{t-1}^{(N)}\}$:

$$\tilde{\omega}_{t-1}^{(i)} = \omega_{t-1}^{(i)} p(y_t|g(x_{t-1}^{(i)}))$$

  - Draw $x_t^{(j)}$ from $p(x_t|x_{t-1}^{(k^j)})$.
  - Compute associated weight

$$\omega_t^{(j)} \propto \frac{p(y_t|x_t^{(j)})}{p(y_t|g(x_{t-1}^{(k^j)}))}.$$

- $\left\{(x_t, \omega_t)^{(i)}\right\}_{i=1}^{N}$ summarizes $p(x_t|y^t)$.

- Maybe add a SIR step to replenish x_ts.

Sample-resample filters

1. Sample $\tilde{x}_{t+1}^{(j)}$ from $q_s(x_{t+1}|x_t^{(j)}, y_{t+1})$;
2. Resample $x_{t+1}^{(i)}$ from $\{\tilde{x}_{t+1}^{(j)}\}_{j=1}^N$ with weights

$$\omega_{t+1}^{(j)} \propto \frac{p(y_{t+1}|\tilde{x}_{t+1}^{(j)})p(\tilde{x}_{t+1}^{(j)}|x_t^{(j)})}{q_s(\tilde{x}_{t+1}^{(j)}|x_t^{(j)}, y_{t+1})}.$$

Bootstrap filter (BF)
BF: $q_s(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|x_t)$ - *blinded sampling*.
BF: $\omega_{t+1} = \omega_t p(y_{t+1}|x_{t+1})$ - *likelihood function*.

Optimal bootstrap filter (OBF)
OBF: $q_s(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|x_t, y_{t+1})$ - *perfectly adapted*.
OBF: $\omega_{t+1} = \omega_t p(y_{t+1}|x_t)$ - *predictive density*.

Resample-sample filters

1. Resample $\tilde{x}_t^{(i)}$ from $\{x_t^{(j)}\}_{j=1}^N$ with weights $q_r(x_t^{(j)}|y_{t+1})$;
2. Sample $x_{t+1}^{(i)}$ from $q_s(x_{t+1}|\tilde{x}_t^{(i)}, y_{t+1})$;
3. New weights

$$\omega_{t+1}^{(i)} = \frac{p(y_{t+1}|x_{t+1}^{(i)})p(x_{t+1}^{(i)}|\tilde{x}_t^{(i)})}{q_r(\tilde{x}_t^{(i)}|y_{t+1})q_s(x_{t+1}^{(i)}|\tilde{x}_t^{(i)}, y_{t+1})}.$$

Auxiliary particle filter (APF)

APF: $q_r(x_t|y_{t+1}) = p(y_{t+1}|g(x_t))$ - *$g(x_t)$ is guess of x_{t+1}*.
APF: $q_s(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|x_t)$ - *blinded sampling*.
APF: $\omega_{t+1} = \omega_t \frac{p(y_{t+1}|x_{t+1})}{p(y_{t+1}|g(\tilde{x}_t))}$ - *likelihood ratio*.

Optimal auxiliary particle filter (OAPF)

OAPF: $q_r(x_t|y_{t+1}) = p(y_{t+1}|x_t)$ - *predictive density*.
OAPF: $q_s(x_{t+1}|x_t, y_{t+1}) = p(x_{t+1}|x_t, y_{t+1})$ - *perfectly adapted*.
OAPF: $\omega_{t+1}^{(i)} = \omega_t^{(i)}$.

Step-by-step filtering

Consider the nonlinear dynamic model (Gordon *et al.*, 1993):

$$\begin{aligned} y_t &\sim N\left(\frac{x_t^2}{20}, 1\right) \\ x_t | x_{t-1} &\sim N(g(x_{t-1}), 10) \end{aligned}$$

where

$$g(x_{t-1}) = 0.5x_{t-1} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8cos(1.2(t-1))$$

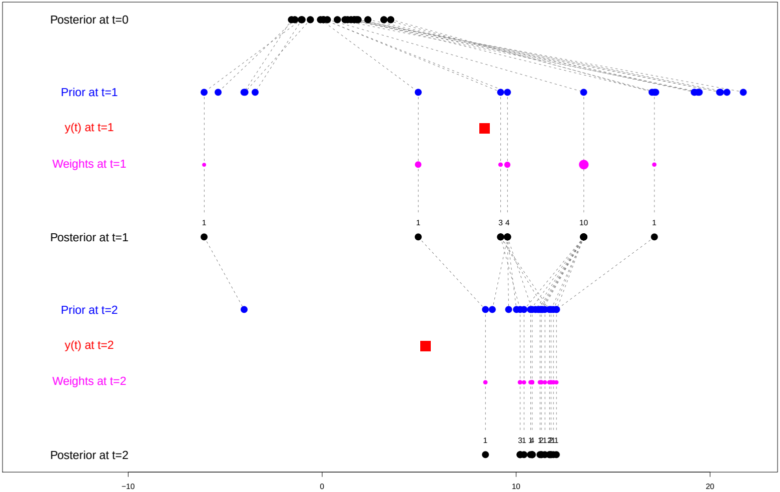
for $t = 1, 2$ and $x_0 = 0.1$.

The two simulated observations are $y_1 = 8.385527$ and 5.336167.

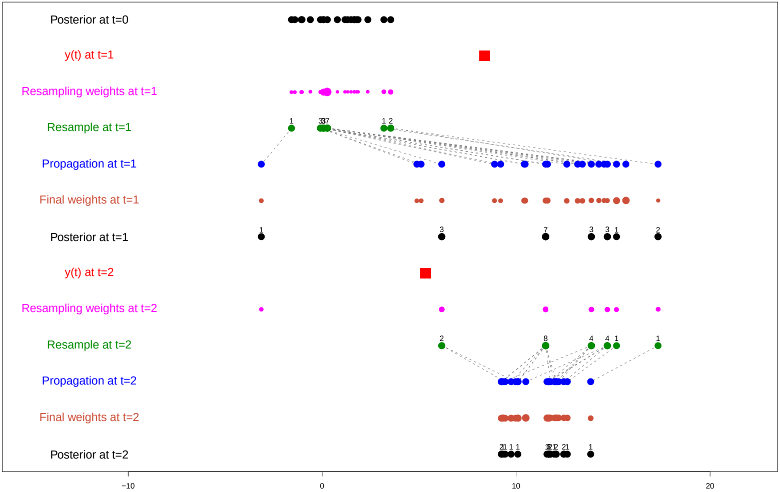
The prior for x_0 is $N(0, 2)$.

BF and APF are run based on $N = 20$ particles.

The bootstrap filter

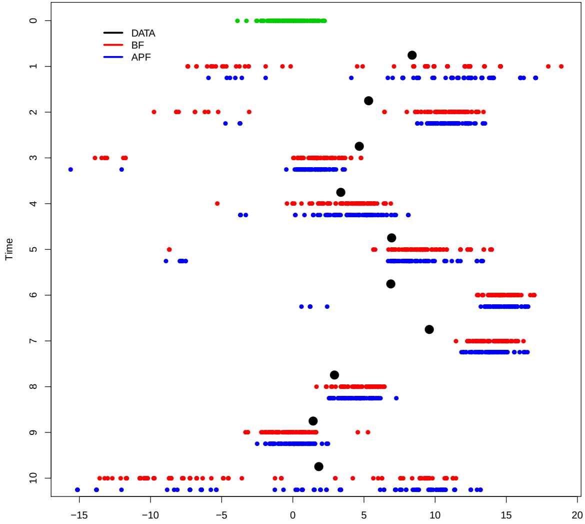


The auxiliary particle filter



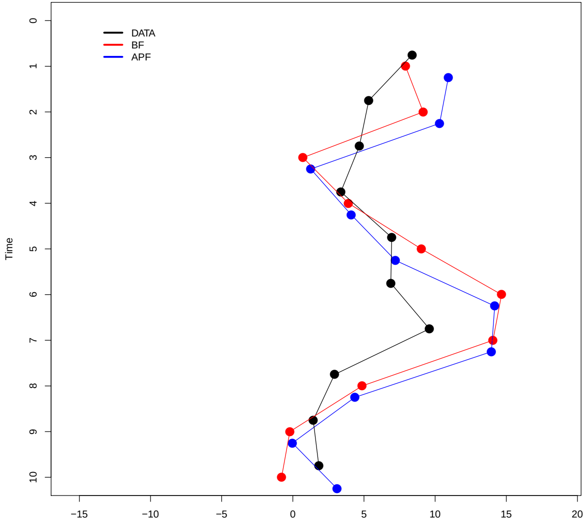
BF and APF

$n = 20$ and $N = 100$.



BF and APF

$n = 20$ and $N = 100$.



Example 2: Simulation exercise

Three data sets ($\tau^2 = (0.25, 0.5, 0.75)$) with $n = 100$ observations were generated from

$$
\begin{aligned}
y_t | x_t &\sim N(x_t, \sigma^2) \\
x_t | x_{t-1} &\sim N(\alpha + \beta x_{t-1}, \tau^2)
\end{aligned}
$$

with $(\alpha, \beta, \sigma^2) = (0.05, 0.95, 1.0)$ and $x_0 = 0.5$.

$x_0 \sim N(0.5, 10)$ and true $p(x_t | y^t)$ are available in closed form.

$R = 20$ replications based on $N = 1000$ particles.

$\text{MAE} = \sum_{t=1}^{T} |\hat{q}_{t,f}^{\alpha} - q_t^{\alpha}| / T$.

where q_t^{α} and $\hat{q}_{t,f}^{\alpha}$ are the true and approximate αth percentile of $p(x_t | y^t)$.

BF, APF, OBF and OAPF

BF is based on $p(x_t|x_{t-1})$ and $p(y_t|x_t)$.

APF is based on $p(x_t|x_{t-1})$ and

$$q_r(x_{t-1}|y_t) \equiv N(\mu_t, \tau^2),$$

where $\mu_t = g(x_{t-1}) = \alpha + \beta x_{t-1}$.

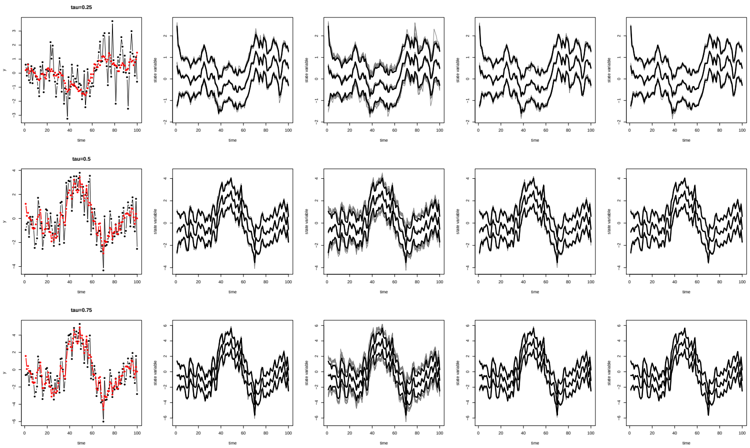
OBF and OAPF are based on

$$\begin{aligned} p(y_t|x_{t-1}) &\equiv N(\mu_t, \sigma^2 + \tau^2) \\ p(x_t|x_{t-1}, y^t) &\equiv N((1-A)\mu_t + Ay_t, A\sigma^2) \end{aligned}$$

where $A = \tau^2/(\sigma^2 + \tau^2)$.

2.5th, 50th and 97.5th percentiles of $p(x_t|y^t)$

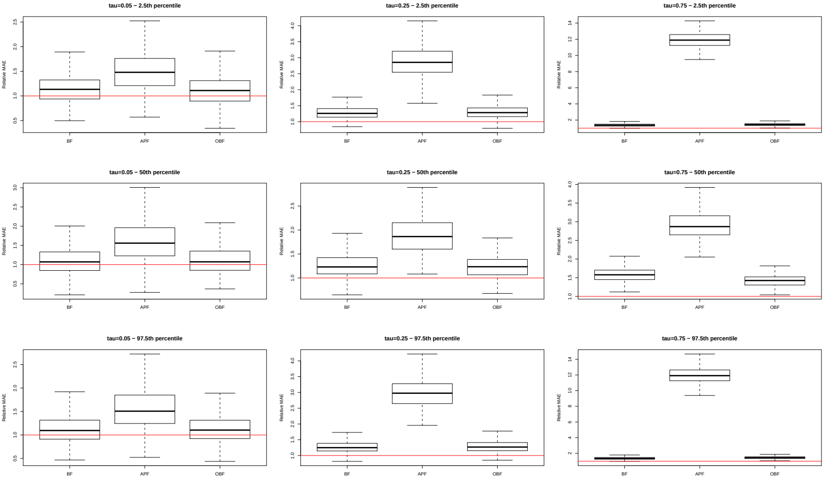
Column 1: y_t (black) versus x_t (red).
Columns 2 and 4: BF and APF (true:black, filter:gray)
Columns 4 and 5: OBF and OAPF (true:black, filter:gray)



Relative MAE

$S = 20$ datasets

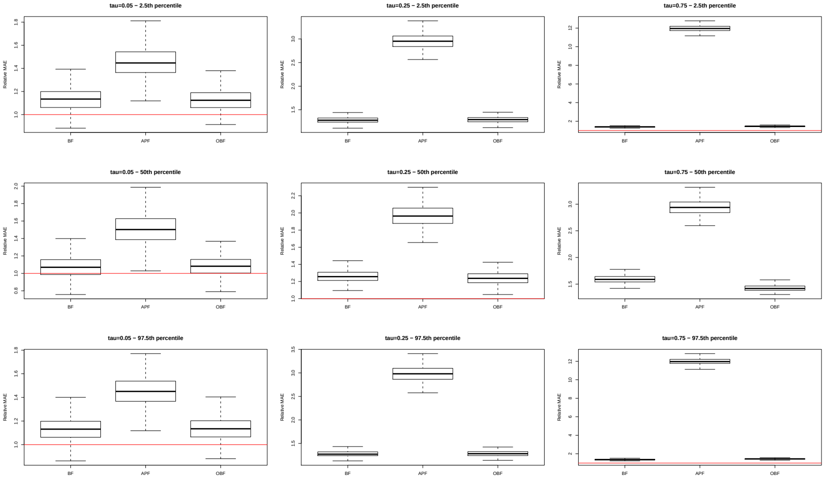
$n = 100$ observations



Relative MAE

$S = 20$ datasets

$n = 1000$ observations



Empirical findings

BF and OBF are similar.

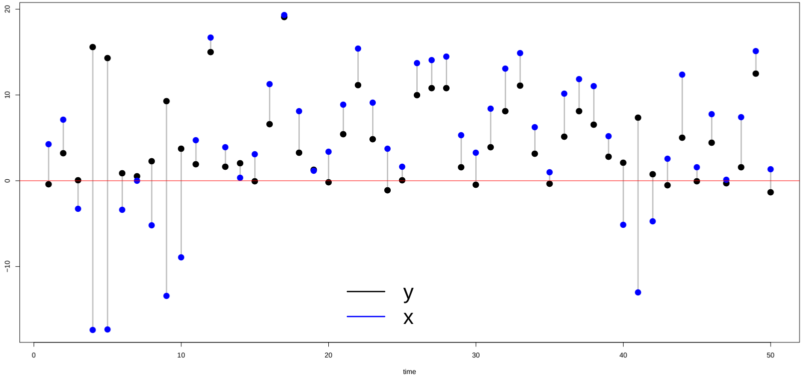
OAPF is significantly better than APF.

OAPF is uniformly better than BF and OBF.

The above findings are more significant when $n = 1000$.

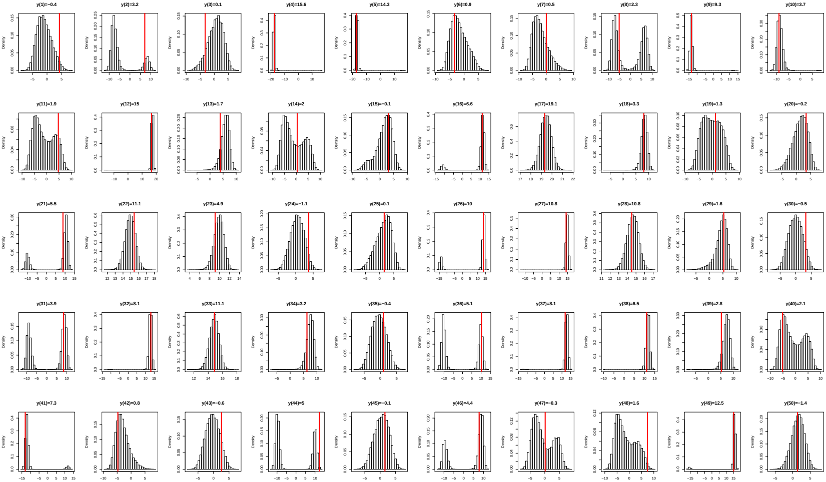
The above findings are more pronounced for larger values of τ^2.

Revisiting the nonlinear dynamic model



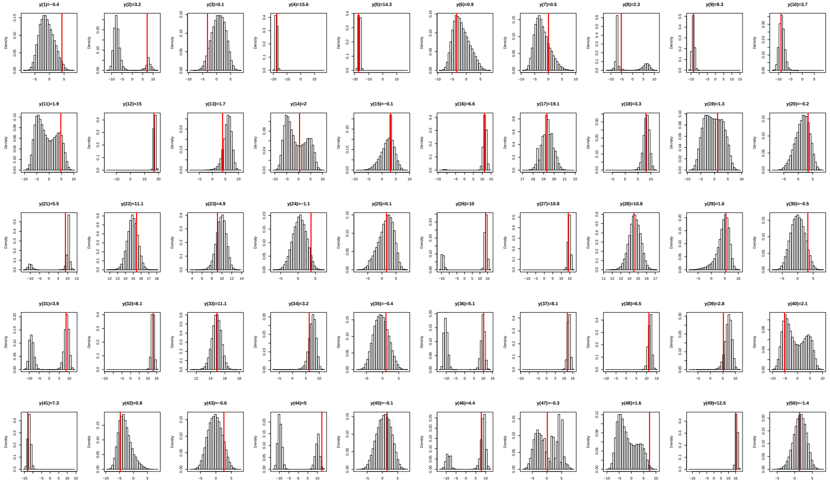
BF:

$\widehat{p}(x_t|y^t)$ for $t = 1, \ldots, n$. $M = 100,000$ particles.



APF:

$\widehat{p}(x_t|y^t)$ for $t = 1, \ldots, n$. $M = 100,000$ particles.



APF's resampling proposal is

$$f_N(y_t; g(x_{t-1}), \sigma^2).$$

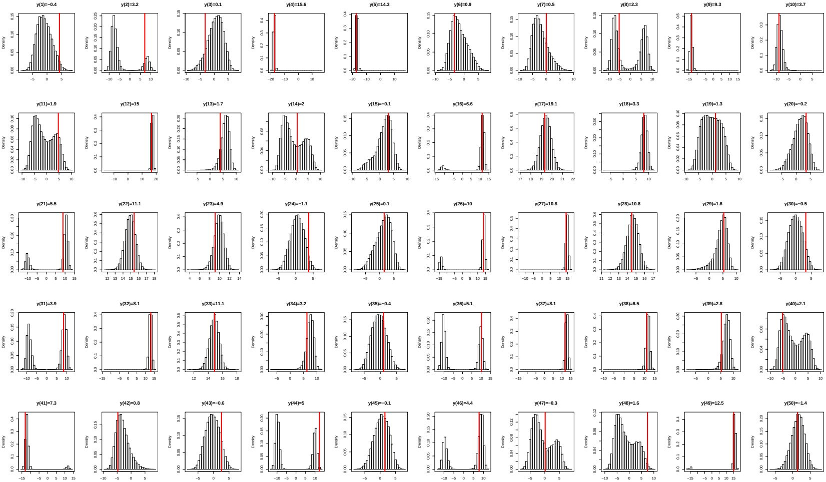
An alternative (potentially better) proposal is

$$f_N(y_t; g(x_{t-1}), \tau^2 g^2(x_{t-1})/100 + \sigma^2),$$

which is based on a 1st order Taylor expansion of $h(x_t) = x_t^2/20$ around $g(x_{t-1})$.

Another APF:

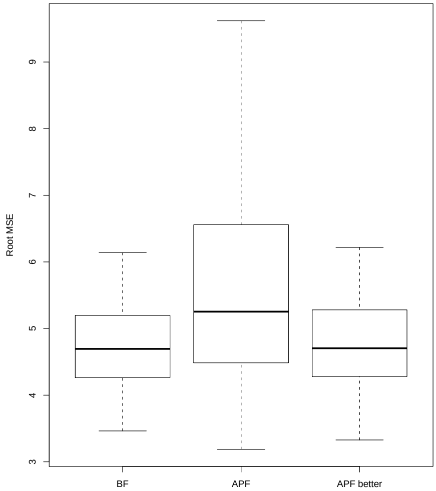
$\widehat{p}(x_t|y^t)\ \forall t.\ \ M = 100,000$ particles.



Root MSE:

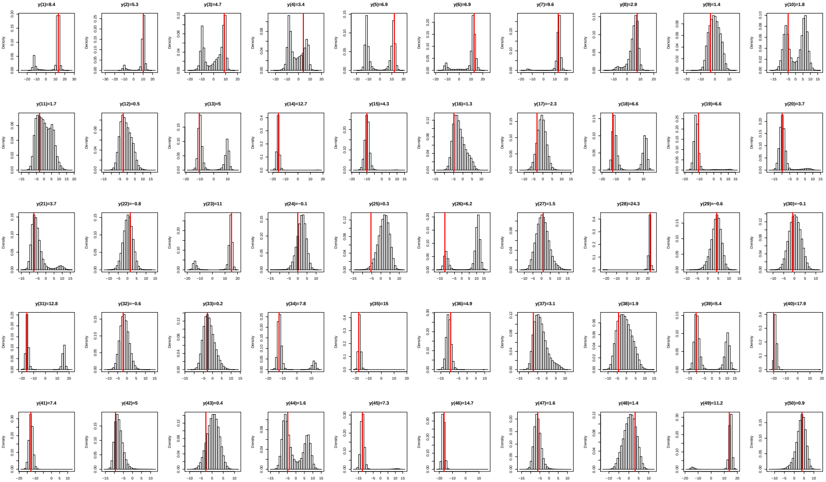
Based on $R = 100$ data sets, $n = 100$ and $M = 1,000$ particles.

Root MSE is $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_t - \widehat{x}_t^f)^2}$, where $\widehat{x}_t^f = \widehat{E}_f(x_t|y^t)$.



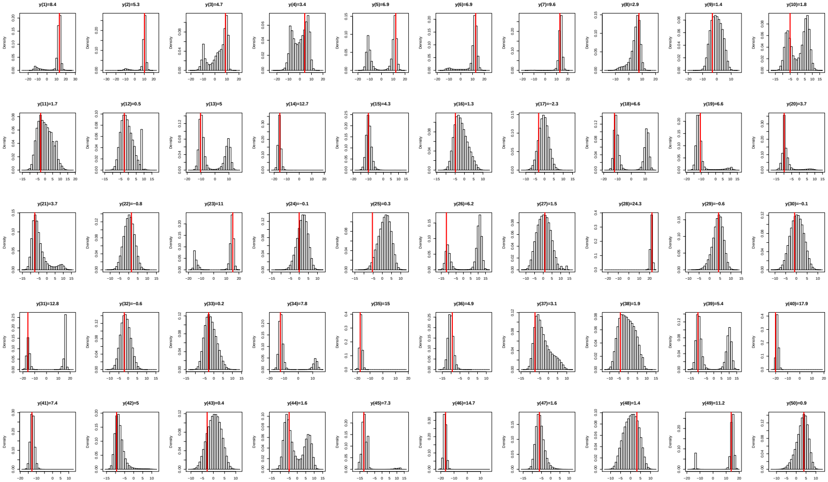
BF + learning (σ^2, τ^2):

$\widehat{p}(x_t | y^t)$ for $t = 1, \ldots, n$. $M = 1{,}000{,}000$ particles.



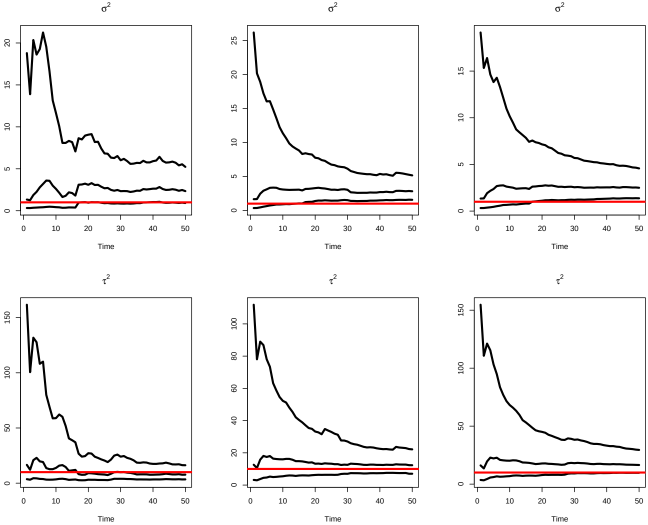
APF + learning (σ^2, τ^2):

$\widehat{p}(x_t|y^t)$ for $t = 1, \ldots, n$. $M = 1{,}000{,}000$ particles.



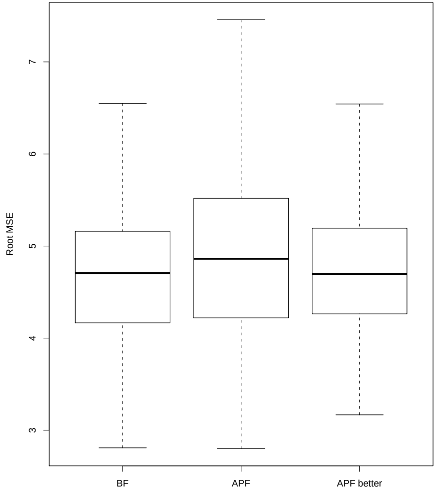
Parameter learning:

$\widehat{p}(\sigma^2|y^t)$ and $p(\tau^2|y^t)$ for $t = 1, \ldots, n$. $M = 1,000,000$ particles.
Left column: BF. *Right column:* APF.



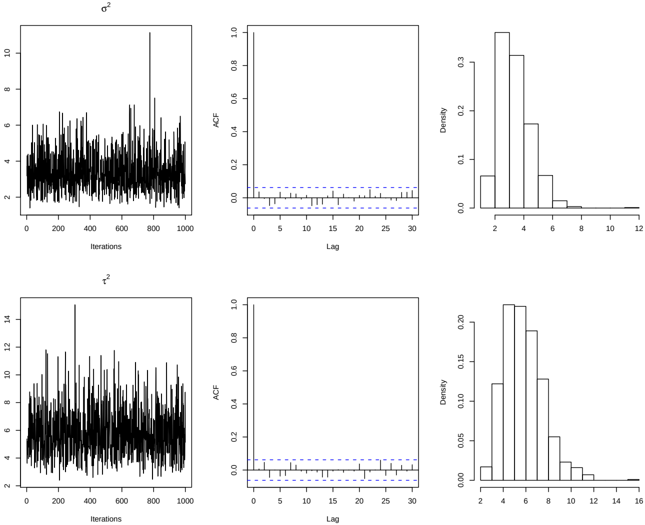
Parameter learning:

Root MSE based on $R = 100$ data sets, $n = 100$ and $M = 1,000$ particles.



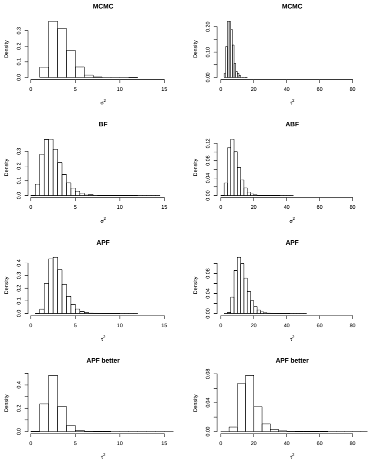
MCMC:

$\widehat{p}(\sigma^2|y^n)$ and $\widehat{p}(\tau^2|y^n)$. Burn-in=$10,000$, Lag=100 and MCMC size=$1,000$.



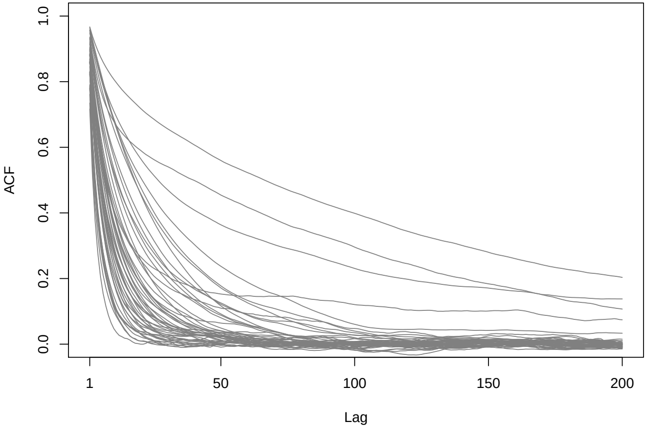
Comparison:

$\widehat{p}(\sigma^2|y^n)$ and $\widehat{p}(\tau^2|y^n)$. MCMC is based on burn-in$=10,000$, Lag$=100$ and MCMC size$=1,000$. Particle filters are based on $M = 1,000,000$ particles.

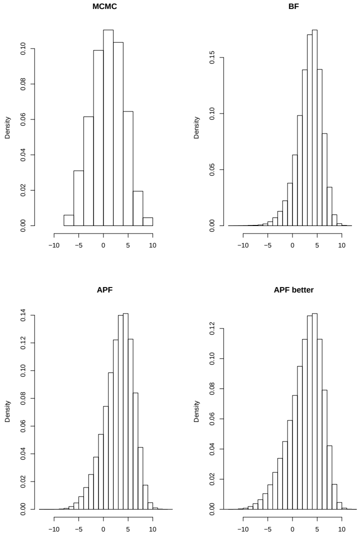


Autocorrelation functions for MCMC draws from $p(x_t|y^n)$.

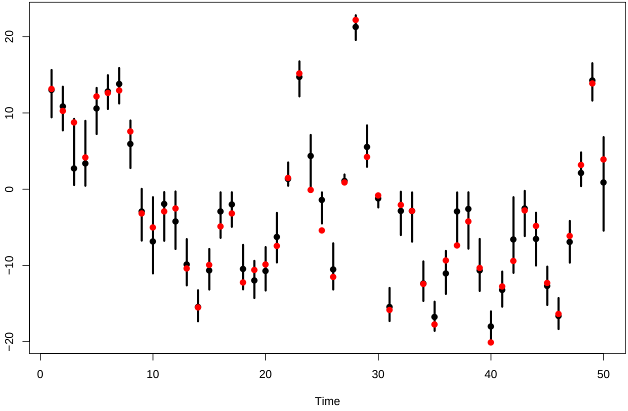
Top graph: based on all $110,000$ draws.
Bottom graph: based on $1,000$ draws (after burn-in$=10,000$ and keeping only 100th draw.



$\widehat{p}(x_n|y^n)$. MCMC is based on burn-in=$10,000$, Lag=100 and MCMC size=$1,000$. Particle filters are based on $M = 1,000,000$ particles.



2.5th, 50th and 97.5th percentiles of $\widehat{p}(x_t|y^n)$ for $t = 1, \ldots, n$. MCMC is based on burn-in=$10,000$, Lag=100 and MCMC size=$1,000$. True values x_ts are the red dots.



Basic references

Gordon, Salmond and Smith (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F 140*, 107-113.

Pitt and Shephard (1999) Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94, 590-599.

Lopes and Tsay (2011) Particle filters and Bayesian inference in financial econometrics, *Journal of Forecasting*, 30, 168-209. `R` code for the examples can be found in
`http://faculty.chicagobooth.edu/hedibert.lopes/research/JForecasting-PF.html`