Particle Learning

Hedibert Freitas Lopes [1]
Booth School of Business
University of Chicago

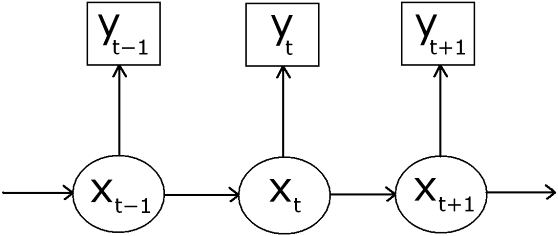
R/Finance 2009
University of Illinois, Chicago, April 25th 2009

[1] Carvalho, Johannes, Lopes and Polson (2008) Particle Learning and Smoothing.

Outline

- Review
  - Sequential learning in Dynamic Linear Models (DLMs)
  - MCMC for DLMs: The FFBS algorithm
  - Sequential importance sampling (SIS)
  - SIS with resampling (SISR)
- Particle Learning (PL)
  - Comparing PL and FFBS
  - Comparing PL and auxiliary particle filter (APF)
  - Comparing Monte Carlo errors
  - Example I: dynamic factor models with time-varying loadings

Toy example: 1st order DLM



The 1st order normal DLM (West and Harrison, 1997) is

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

where $x_0 \sim N(m_0, C_0)$ and $\theta = (\sigma^2, \tau^2)$ fixed (for now).

Toy example: sequential Bayes learning

Evolution-prediction-updating

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

where $y^t = (y_1, \ldots, y_t)$.

- Posterior at t: $(x_t|y^t) \sim N(m_t, C_t)$
- Prior at $t+1$: $(x_{t+1}|y^t) \sim N(m_t, R_{t+1})$
- Marginal likelihood: $(y_{t+1}|y^t) \sim N(m_t, Q_{t+1})$
- Posterior at $t+1$: $(x_{t+1}|y^{t+1}) \sim N(m_{t+1}, C_{t+1})$

where $R_{t+1} = C_t + \tau^2$, $Q_{t+1} = R_{t+1} + \sigma^2$, $A_{t+1} = R_{t+1}/Q_{t+1}$, $C_{t+1} = A_{t+1}\sigma^2$, and $m_{t+1} = (1 - A_{t+1})m_t + A_{t+1}y_{t+1}$.

Toy example: backward smoothing

For $t = n$, $x_n | y^n \sim N(m_n^n, C_n^n)$, where

$$\begin{aligned} m_n^n &= m_n \\ C_n^n &= C_n \end{aligned}$$

For $t < n$, $x_t | y^n \sim N(m_t^n, C_t^n)$, where

$$\begin{aligned} m_t^n &= (1 - B_t) m_t + B_t m_{t+1}^n \\ C_t^n &= (1 - B_t) C_t + B_t^2 C_{t+1}^n \end{aligned}$$

and

$$B_t = \frac{C_t}{C_t + \tau^2}$$

Toy example: backward sampling

For $t = n$, $x_n | y^n \sim N(a_n^n, R_n^n)$, where

$$\begin{aligned} a_n^n &= m_n \\ R_n^n &= C_n \end{aligned}$$

For $t < n$, $x_t | x_{t+1}, y^n \sim N(a_t^n, R_t^n)$, where

$$\begin{aligned} a_t^n &= (1 - B_t) m_t + B_t x_{t+1} \\ R_t^n &= B_t \tau^2 \end{aligned}$$

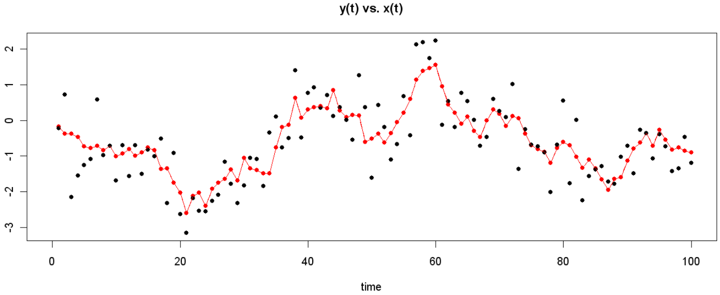
and

$$B_t = \frac{C_t}{C_t + \tau^2}$$

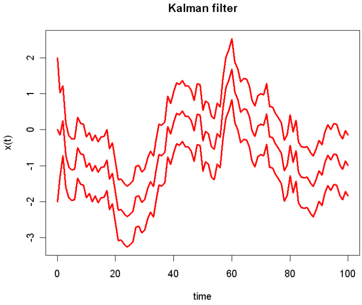
This is basically the Forward filtering, backward sampling algorithm used to sample from $p(x^n | y^n)$ (Carter and Kohn, 1994 and Frühwirth-Schnatter, 1994).

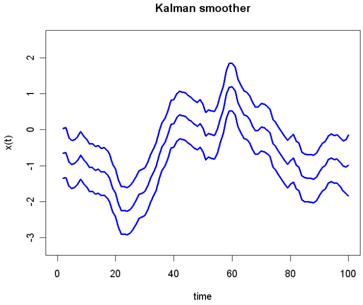
Toy example: simulated data

$T = 100$, $\tau^2 = 1.0$, $\sigma^2 = 0.5$, $m_0 = 0$ and $C_0 = 1$.



Toy example: $p(x_t|y^t)$ versus $p(x_t|y^n)$





Toy example: integrating out states x^n

We showed earlier that

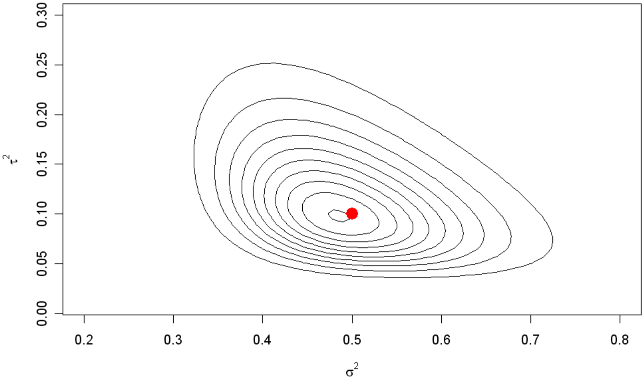
$$(y_t|y^{t-1}) \sim N(m_{t-1}, Q_t)$$

where both m_{t-1} and Q_t were presented before and are functions of $\theta = (\sigma^2, \tau^2)$, y^{t-1}, m_0 and C_0.

Therefore, by Bayes' rule,

$$\begin{aligned} p(\theta|y^n) &\propto p(\theta)p(y^n|\theta) \\ &= p(\theta)\prod_{t=1}^{n} f_N(y_t|m_{t-1}, Q_t). \end{aligned}$$

Toy example: $p(y|\sigma^2, \tau^2)$



Practically untractable, either analytically or numerically, when the dimension θ is moderately large, say larger than 10.

Toy example: MCMC scheme

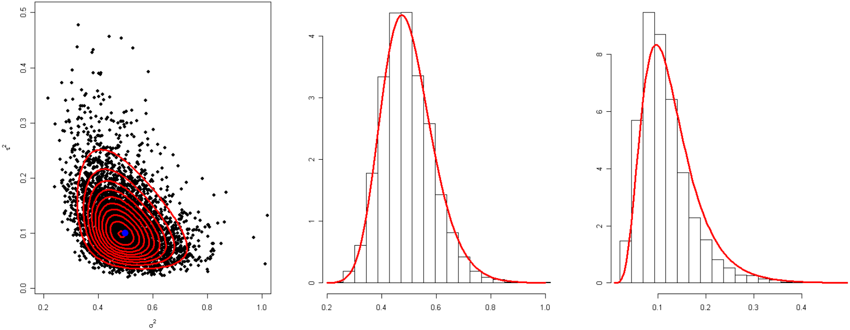
- Sample θ from $p(\theta|y^n, x^n)$

$$p(\theta|y^n, x^n) \propto p(\theta) \prod_{t=1}^{n} p(y_t|x_t, \theta) p(x_t|x_{t-1}, \theta).$$

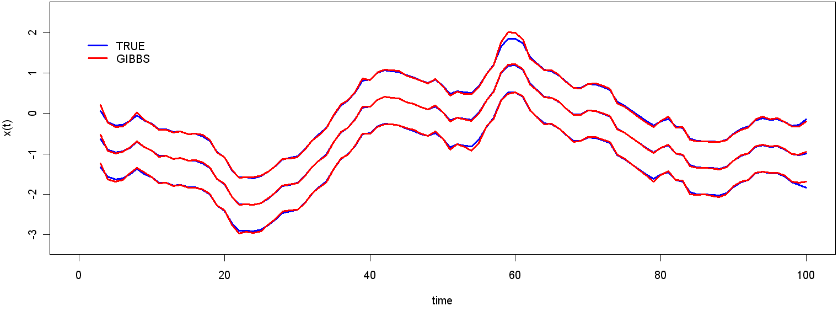
- Sample x^n from $p(x^n|y^n, \theta)$

$$p(x^n|y^n, \theta) = \prod_{t=1}^{n} f_N(x_t|a_t^n, R_t^n)$$

Toy example: $p(\sigma^2, \tau^2 | y^n)$



Toy example: $p(x_t | y^n)$



Lessons from toy example

Sequential learning in non-normal and nonlinear dynamic models $p(y_{t+1}|x_{t+1})$ and $p(x_{t+1}|x_t)$ in general rather difficult since

$$
\begin{aligned}
p(x_{t+1}|y^t) &= \int p(x_{t+1}|x_t)p(x_t|y^t)dx_t \\
p(x_{t+1}|y^{t+1}) &\propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y^t)
\end{aligned}
$$

are usually unavailable in closed form.

Over the last 20 years:

- FFBS for conditionally Gaussian DLMs;
- Gamerman (1998) for generalized DLMs;
- Carlin, Polson and Stoffer (2002) for more general DMs.

Sequential importance sampling (SIS) filter [2]

- Posterior at t:

$$\{x_t^{(i)}, \omega_t^{(i)}\}_{i=1}^N \sim p(x_t|y^t)$$

- Propagation:

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

- New weights:

$$\omega_{t+1}^{(i)} \propto \omega_t^{(i)} p(y_{t+1}|x_{t+1}^{(i)})$$

- Posterior at $t+1$:

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

- Maybe add a resampling step.

[2] Smith and Gelfand (1992); Kong, Liu and Wong (1994)

SIS with Resampling (SISR)

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

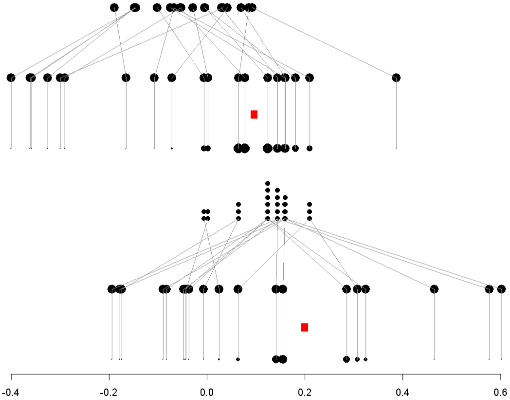
$$\tilde{x}_{t+1}^i \sim p(x_{t+1}|x_t^i)$$

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

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

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

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



Uniform weights is the goal!

Liu's (1996) effective sample size

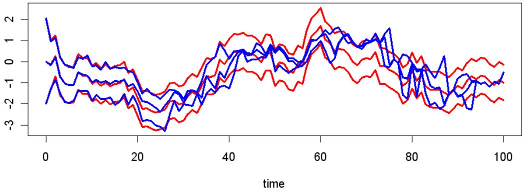
Let ω_t be the normalized weights. Then,

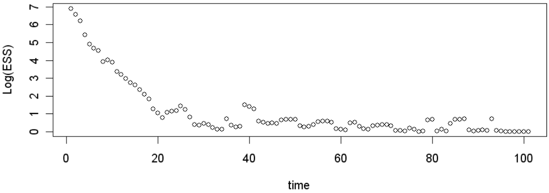
$$ESS_t = \frac{1}{\sum_{i=1}^{N} \left(\omega_t^{(i)}\right)^2}$$

If the $ESS = N$, the weights are equally balanced.

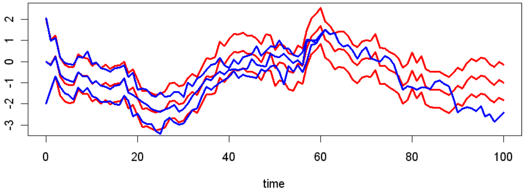
If the $ESS \approx 1$, then *particle degeneracy*.

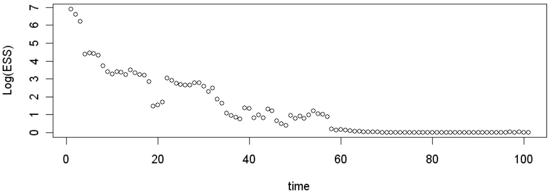
Toy example (revisited): SIS filter, $M = 1000$



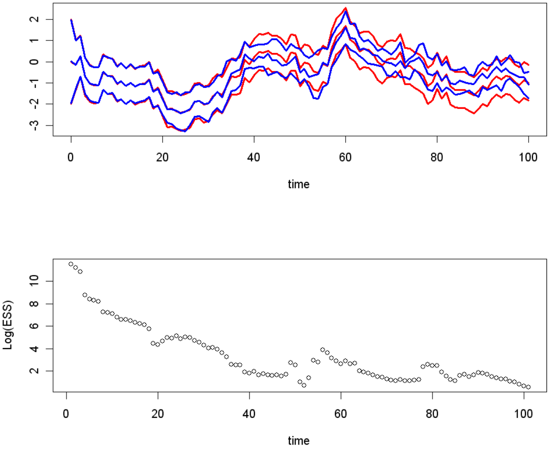


Toy example: SISR filter, $M = 1000$





Toy example: SISR filter, $M = 100000$



Particle learning (PL)

- Advantages:
  - Sequentially learning about (x_t, θ);
  - A real/practical alternative to MCMC methods;
  - Sequential model assessment;
  - Applicable in a wide range of dynamic and static models.

- Key issues in our approach:
  - Reverse the Kalman filter logic: resample/propagate;
  - Estimation of "fixed", unknown parameters;
  - Work with conditional sufficient statistics;
  - Use SMC for smoothing.

Parameter Learning[3]

It is assumed that

$$p\left(\theta|x^t, y^t\right) = p\left(\theta|s_t\right),$$

where s_t is a recursively defined sufficient statistic (SS),

$$s_{t+1} = \mathcal{S}\left(s_t, x_{t+1}, y_{t+1}\right).$$

- SS are just another state with a deterministic evolution;
- "Filtering" SS provides a mechanism for replenishing the parameters avoiding degeneracy;
- Reduction of the variance of re-sampling weights: Rao-Blackwellization.

[3] Liu and West (2001), Storvik (2002) and Fearnhead (2002)

Reverse the Kalman filter logic

Traditional logic for filtering (Kalman, 1960): Predict/Update

$$p\left(x_{t+1}|y^{t}\right)=\int p\left(x_{t+1}|x_{t}\right)p\left(x_{t}|y^{t}\right)dx_{t}$$

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

By Bayes rule, we can reverse this logic through:

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

$$p\left(x_{t+1}|y^{t+1}\right)=\int p\left(x_{t+1}|x_{t},y_{t+1}\right)p\left(x_{t}|y^{t+1}\right)dx_{t}$$

• We will first re-sample (smooth) and then propagate. Since information in y_{t+1} is used in both steps, the algorithm will be more efficient.

The General Approach

- Assume at time t, $\left\{(x_t, s_t)^{(i)}\right\}_{i=1}^{N}$ approximates $p^N(x_t, s_t|y^t)$;

- Once y_{t+1} is observed, the re-sample/propagation rule is

  - Resampling

$$p\left(x_t, s_t|y^{t+1}\right) \propto p\left(y_{t+1}|x_t, s_t\right) p\left(x_t, s_t|y^t\right)$$

  - Propagation

$$p\left(x_{t+1}|y^{t+1}\right) = \int p\left(x_{t+1}|x_t, s_t, y_{t+1}\right) p\left(x_t, s_t|y^{t+1}\right) dx_t ds_t$$

Discussion

Target: $p\left(x_{t+1}, x_t | y^{t+1}\right)$

IS weights:

$$w \propto \frac{p\left(x_{t+1}|x_t, y_{t+1}\right) p\left(y_{t+1}|x_t\right) p\left(x_t|y^t\right)}{q_1\left(x_t|y_{t+1}\right) q_2\left(x_{t+1}|x_t, y_{t+1}\right)}$$

$$w \propto \frac{p\left(x_{t+1}|x_t, y_{t+1}\right) p\left(y_{t+1}|x_t\right) p\left(x_t|y^t\right)}{p\left(x_t|y_{t+1}\right) p\left(x_{t+1}|x_t, y_{t+1}\right) p\left(x_t|y^t\right)} = 1 \quad \text{PL: Exact Filter}$$

- This is the ideal scenario and should serve as a guiding principle in the construction of SeqMC filters [4].

- Marginalization is key!

[4] "Perfect adaptation" discussed in Pitt and Shephard (1999) and Johansen and Doucet (2008).

Toy example: PL in action

- Posterior at t: $\Phi_t \equiv \left\{x_t^{(i)}\right\}_{i=1}^N \sim p(x_t|y^t)$.
- Compute, for $i = 1, \ldots, N$,

$$w_{t+1}^{(i)} \propto p(y_{t+1}|x_t^{(i)}) \equiv f_N(y_{t+1}; x_t, \sigma^2)$$

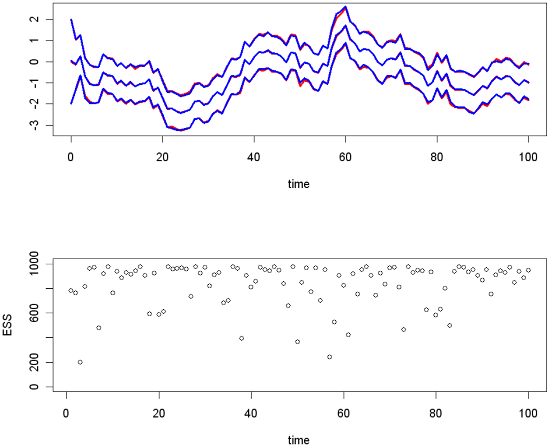
- Resample from Φ_t with weights w_{t+1}: $\tilde{\Phi}_t \equiv \left\{\tilde{x}_t^{(i)}\right\}_{i=1}^N$.
- Propagate states, for $i = 1, \ldots, N$,

$$x_{t+1}^{(i)} \sim p(x_{t+1}|\tilde{x}_t^{(i)}, y_{t+1}) \equiv f_N(x_{t+1}; b, B)$$

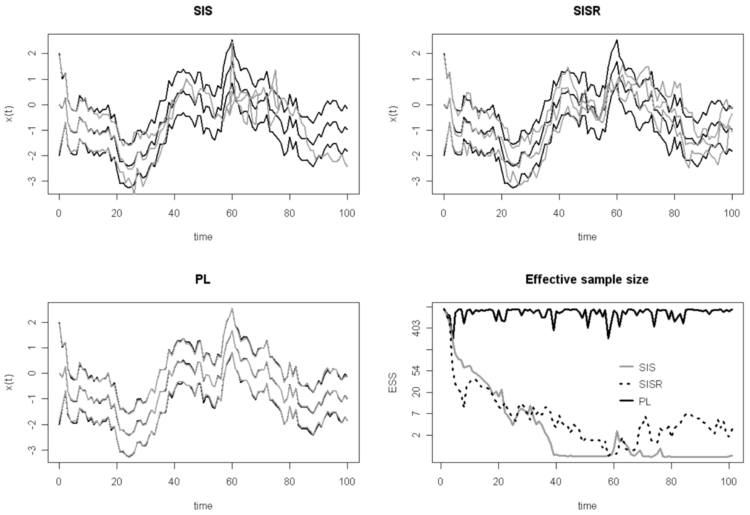
where $B^{-1} = \tau^{-2} + \sigma^{-2}$ and $b = B(\sigma^{-2} y_{t+1} + \tau^{-2} x_t)$.

- Posterior at $t+1$: $\Phi_{t+1} \equiv \left\{x_{t+1}^{(i)}\right\}_{i=1}^N \sim p(x_{t+1}|y^{t+1})$.

Toy example: PL pure filter, $M = 1000$



Toy example: comparing SIS, SISR and PL



Smoothing via PL[5]

After filtering, smoothing is straightforward:

$$p\left(x^T, \theta | y^T\right) = p(\theta | y^T) p\left(x_T | y^T, \theta\right) \prod_{t=T-1}^{1} p\left(x_t | x_{t+1}, y^{t+1}, \theta\right);$$

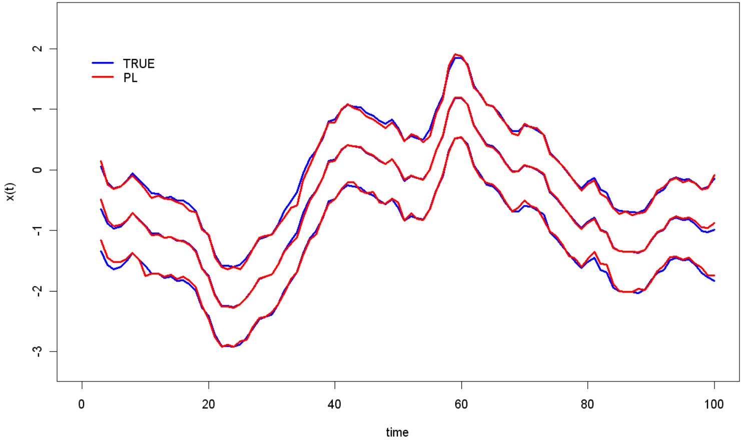
where (by Bayes rule and conditional independence)

$$p\left(x_t | x_{t+1}, y^{t+1}, \theta\right) \quad \propto \quad p\left(x_{t+1} | x_t, \theta\right)$$

As we will discuss, this can be a very attractive alternative to MCMC in state-space models.

[5]Extension of Godsill, Doucet and West (2004).

Toy example: smoothing



Studying Monte Carlo variation

Three time series of length $T = 1000$ were simulated from

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

with $x_0 = 0$ and (σ^2, τ^2) in $\{(0.1, 0.01), (0.01, 0.01), (0.01, 0.1)\}$. Throughout σ^2 is kept fixed.

The independent prior distributions for x_0 and τ^2 are $x_0 \sim N(m_0, V_0)$ and $\tau^2 \sim IG(a, b)$, for $a = 10$, $b = (a+1)\tau_0^2$, $m_0 = 0$ and $V_0 = 1$, where τ_0^2 is the true value of τ^2 for a given study.

We compared SISR, propagation-resampling and PL.

In all filters τ^2 is sampled offline from $p(\tau^2 | S_t)$ where S_t is the vector of conditional sufficient statistics.

Comparing filters via MAE

The three filters are rerun $R = 100$ times, all with the same seed within run, for each one of the three simulated data sets. Five different number of particles N were considered: 250, 500, 1000, 2000 and 5000.

Mean absolute errors (MAE) taken over the 100 replications are constructed by comparing percentiles of the true sequential distributions $p(x_t|y^t)$ and $p(\tau^2|y^t)$ to percentiles of the estimated sequential distributions $p_N(x_t|y^t)$ and $p_N(\tau^2|y^t)$.

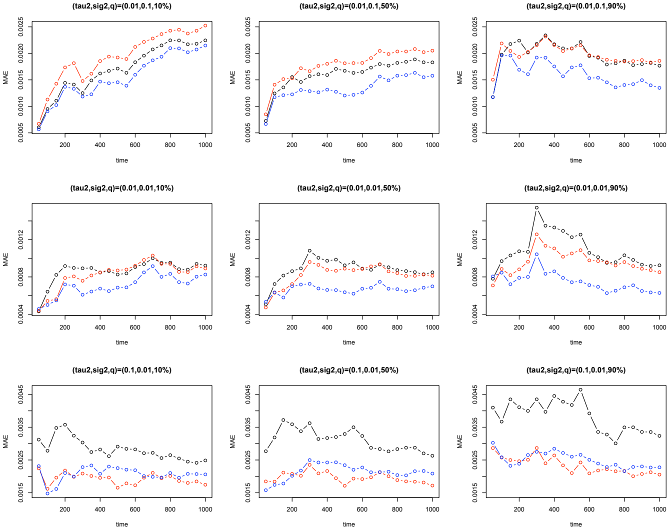
For $\alpha = 0.1, 0.5, 0.9$, true and estimated values of $q^x_{t,\alpha}$ and $q^{\tau^2}_{t,\alpha}$ were computed, for $Pr(x_t < q^x_{t,\alpha}|y^t) = Pr(\tau^2 < q^{\tau^2}_{t,\alpha}|y^t) = \alpha$.

For a in $\{x, \tau^2\}$ and α in $\{0.01, 0.50, 0.99\}$,

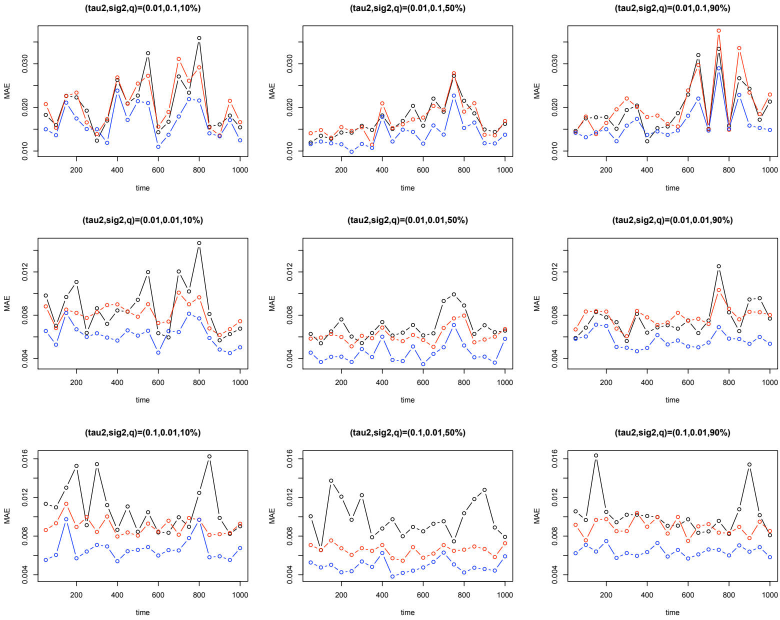
$$MAE^a_{t,\alpha} = \frac{1}{R}\sum_{r=1}^{R} |q^a_{t,\alpha} - \hat{q}^a_{t,\alpha,r}|$$

$M = 500$ and learning τ^2.

SISR,propagate-resample,PL.



$M = 500$ and learning x_t.



General PL

- Posterior at t: $\Phi_t \equiv \left\{(x_t, \theta)^{(i)}\right\}_{i=1}^{N} \sim p(x_t, \theta | y^t)$.
- Compute, for $i = 1, \ldots, N$,

$$w_{t+1}^{(i)} \propto p(y_{t+1} | x_t^{(i)}, \theta^{(i)})$$

- Resample from Φ_t with weights w_{t+1}: $\tilde{\Phi}_t \equiv \left\{(\tilde{x}_t, \tilde{\theta})^{(i)}\right\}_{i=1}^{N}$.
- For $i = 1, \ldots, N$,
  - Propagate states

$$x_{t+1}^{(i)} \sim p(x_{t+1} | \tilde{x}_t^{(i)}, \tilde{\theta}^{(i)}, y_{t+1})$$

  - Update sufficient statistics

$$s_{t+1}^{(i)} = \mathcal{S}(s_t^{(i)}, x_{t+1}^{(i)}, y_{t+1})$$

  - Sample parameters

$$\theta^{(i)} \sim p(\theta | s_{t+1}^{(i)})$$

Example: dynamic factor model with switching loadings [6]

For $t = 1, \ldots, T$, the model is defined as follows:

- Observation equation

$$y_t | z_t, \theta \sim N(\gamma_t x_t, \sigma^2 I_2)$$

- State equations

$$\begin{aligned} x_t | x_{t-1}, \theta &\sim N(x_{t-1}, \sigma_x^2) \\ \lambda_t | \lambda_{t-1}, \theta &\sim Ber((1-p)^{1-\lambda_{t-1}} q^{\lambda_{t-1}}) \end{aligned}$$

where $z_t = (x_t, \lambda_t)'$, $\gamma_t = (1, \beta_{\lambda_t})'$ is the vector of time-varying loadings and $\theta = (\beta_1, \beta_2, \sigma^2, \sigma_x^2, p, q)'$ is the vector of fixed parameters.

γ are the factor loadings and x_t the common dynamic factors.

[6] Lopes and Carvalho (2007) and Lopes, Salazar and Gamerman (2008)

Prior information

The prior distributions are conditionally conjugate:

$$
\begin{aligned}
(\beta_i|\sigma^2) &\sim N\left(b_{i0}, \sigma^2 B_{i0}\right) \qquad \text{for } i = 1, 2, \\
\sigma^2 &\sim IG\left(\frac{\nu_{00}}{2}, \frac{d_{00}}{2}\right) \\
\sigma_x^2 &\sim IG\left(\frac{\nu_{10}}{2}, \frac{d_{10}}{2}\right) \\
p &\sim Beta(p_1, p_2) \\
q &\sim Beta(q_1, q_2) \\
x_0 &\sim N(m_0, C_0)
\end{aligned}
$$

Particle representation

At time t, particles

$$\left\{ (x_t, \lambda_t, \theta, s_t^x, s_t)^{(i)} \right\}_{i=1}^{N}$$

approximating

$$p\left(x_t, \lambda_t, \theta, s_t^x, s_t | y^t\right)$$

where

- $s_t^x = \mathcal{S}(s_{t-1}^x, \theta)$ are state sufficient statistics
- $s_t = \mathcal{S}(s_{t-1}, x_t, \lambda_t)$ are fixed parameter sufficient statistics

Re-sampling $(x_t, \lambda_t, \theta, s_t^x, s_t)$

Let us redefine $\beta_i = (1, \beta_i)'$ whenever necessary.

Draw an index $k(i) \sim Multi(\omega^{(i)})$ with weights

$$\omega^{(i)} \propto p(y_{t+1} | (s_t^x, \lambda_t, \theta)^{k(i)})$$

with

$$p(y_{t+1} | m_t, C_t, \lambda_t, \theta) = \sum_{j=1}^{2} f_N \left(y_{t+1}; \beta_j m_t, V_j \right) Pr \left(\lambda_{t+1} = j | \lambda_t, \theta \right)$$

where $V_j = (C_t + \sigma_x^2)\beta_j\beta_j' + \sigma^2 I_2$, m_t and C_t are components of s_t^x and f_N denotes the normal density function.

Propagating states

- Draw auxiliary state λ_{t+1}

$$\lambda_{t+1}^{(i)} \sim p(\lambda_{t+1} | (s_t^x, \lambda_t, \theta)^{k(i)}, y_{t+1})$$

where

$$Pr(\lambda_{t+1} = j | s_t^x, \lambda_t, \theta, y_{t+1}) \propto f_N\left(y_{t+1}; \beta_j m_t, V_j\right) p\left(\lambda_{t+1} = j | \lambda_t, \theta\right).$$

- Draw state x_{t+1} conditionally on λ_{t+1}

$$x_{t+1}^{(i)} \sim p(x_{t+1} | \lambda_{t+1}^{(i)}, (s_t^x, \theta)^{k(i)}, y_{t+1})$$

by a simply Kalman filter update.

Updating states sufficient statistics, s^x_{t+1}

The Kalman filter recursion yield

$$
\begin{aligned}
m_{t+1} &= m_t + A_{t+1}(y_{t+1} - \beta_{\lambda_{t+1}} m_t) \\
C_{t+1} &= C_t + \sigma_x^2 - A_{t+1} Q_{t+1}^{-1} A'_{t+1}
\end{aligned}
$$

where

$$
\begin{aligned}
Q_{t+1} &= (C_t + \sigma_x^2)\gamma_{t+1}\gamma'_{t+1} + \sigma^2 I_2 \\
A_{t+1} &= (C_t + \sigma_x^2)\gamma'_{t+1} Q_{t+1}^{-1}
\end{aligned}
$$

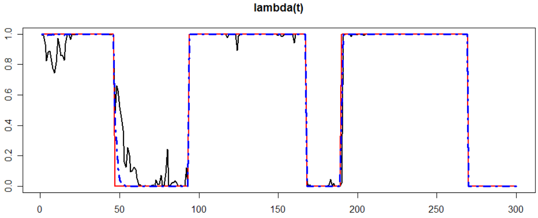
Updating parameter sufficient statistics, s_{t+1}

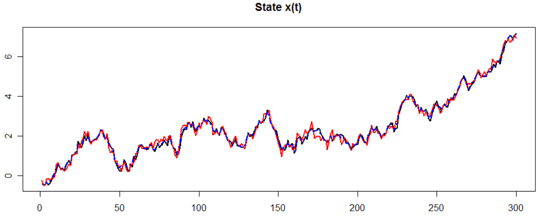
Recall that $s_{t+1} = \mathcal{S}(s_t, x_{t+1}, \lambda_{t+1})$. Then,

$$
\begin{aligned}
(\beta_i|\sigma^2, s_{t+1}) &\sim N\left(b_{i,t+1}, \sigma^2 B_{i,t+1}\right) \qquad \text{for } i = 1, 2, \\
(\sigma^2|s_{t+1}) &\sim IG\left(\frac{\nu_{0t}}{2}, \frac{d_{0,t+1}}{2}\right) \\
(\sigma_x^2|s_{t+1}) &\sim IG\left(\frac{\nu_{1t}}{2}, \frac{d_{1,t+1}}{2}\right) \\
(p|s_{t+1}) &\sim Beta(p_{1,t+1}, p_{2,t+1}) \\
(q|s_{t+1}) &\sim Beta(q_{1,t+1}, q_{2,t+1})
\end{aligned}
$$

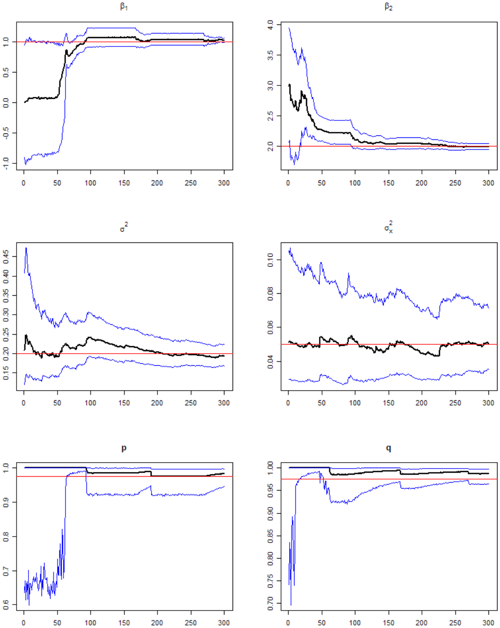
where $\mathbb{I}_{\lambda_{t+1}=i} = \mathbb{I}_i$, $\mathbb{I}_{\lambda_t=i,\lambda_{t+1}=j} = \mathbb{I}_{ij}$, $\nu_{it} = \nu_{i,t-1} + 1$,
$B_{i,t+1}^{-1} = B_{it}^{-1} + x_{t+1}^2$, $B_{i,t+1}^{-1} b_{i,t+1} = B_{it}^{-1} b_{it} + x_{t+1} y_{t+1,2} \mathbb{I}_i$,
$p_{i,t+1} = p_{it} + \mathbb{I}_{1i}$ (similarly for $q_{i,t+1}$) for $i = 1, 2$,
$d_{0,t+1} = d_{0,t} + \left(y_{t+1,1} - x_{t+1}\right)^2 +$
$\sum_{j=1}^{2} \left[\left(y_{t+1,2} - b_{j,t+1} x_{t+1}\right) y_{t+1,2} + B_{j,t+1}^{-1} b_{j,t+1}\right] \mathbb{I}_j$, and
$d_{1,t+1} = d_{1,t} + \left(x_{t+1} - x_t\right)^2$.

States filtering and smoothing





Sequential parameter learning



Work in progress

1. Particle Learning and Smoothing (CaJLoPo)
2. PL in General Mixtures (CaLoPoT)
3. PL for Generalized Conditional DLM (CaLoPo)
4. Comparing sequential Monte Carlo filters (CaJLoPo)
5. Bayesian statistics with a smile (PoLoCa)
6. PL for Autoregressive Models with Structured Priors (PrLo)
7. PL in Epidemic SEIR Models (DLoPo)
8. PL Without Conditional Sufficient Statistics (NCaLo)
9. PL for Long Memory Stochastic Volatility Models (MaLo)
10. PL for DSGE Models (PeChCaLo)
11. Stochastic Volatility Shot-Noise (CaJLoPo)
12. Options, SV and Jumps in the Interest Rate Risk Premia (LuLo)

Ca: Carvalho; Ch: Chen; D: Dukic; J: Johannes; Lo: Lopes; Lu: Lund;
Ma: Macaro; N: Jarad Niemi; Pe: Petralia; Po: Polson; Pr: Prado; T: Taddy.