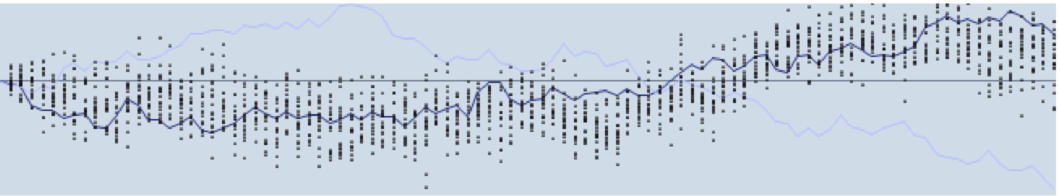
Sequential Monte Carlo Methods in R



Thomas Jakobsen
Jeffrey Todd Lins
Saxo Bank A/S

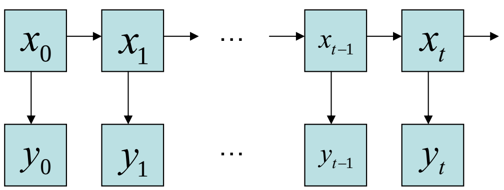
jtl@saxobank.com, tj@saxobank.com

Overview

- The model
- An introduction to sequential Monte Carlo methods
- Algorithm & implementation
- A factor stochastic volatility model
- Example, artificial 2-factor model
- Analysis of forex data
- Conclusions

The Model

- Markovian, nonlinear, non-Gaussian state-space model:



X: Unobserved variables
Y: Observations
θ: Parameters

- Described by

$p(x_0)$		Prior state distribution
$p(x_t \mid x_{t-1}, \Theta)$	for $t \geq 1$	Markovian state space model
$p(y_t \mid x_t, \Theta)$	for $t \geq 1$	Observational model

- Observations arrive sequentially and are noisy.
- Problem statement:
  - Estimate recursively in time the posterior distribution p(x|y,Θ). ("tracking the state")
  - Additionally: Estimate Θ.

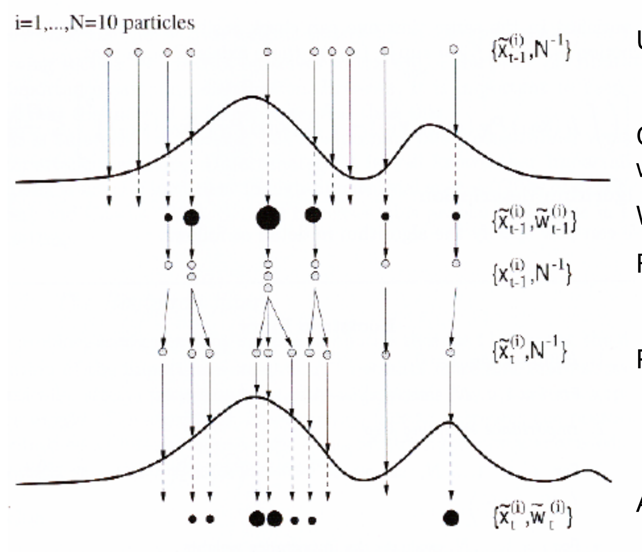
Sequential Monte Carlo Methods

- Useful when a (partially observed) state needs to be tracked or forecasted:
  - Tracking problems (robots, vision, radar etc.)
  - Time series analysis (economical/financial data etc.)
  - General online inference
- Sequential Monte Carlo methods are algorithms for inference in hidden state space models.
- Also known as particle filters, condensation, sampling importance resampling etc.

Sequential Monte Carlo Methods

- SMC methods: Basically a nonlinear, non-Gaussian version of the Kalman filter (but approximate – not closed form)
- The posterior at time t-1 is represented by a set of weighted particles. The particles are drawn i.i.d. and recursively updated.
- Next slide: Illustration of update

A bootstrap approach (from [1])



Unweighted measure

Compute importance weights using info at time t-1

Weighted measure

Resampling

Prediction

Apprx. of p(x_t|y_1:t-1)

Comb. Parameter and State Estimation

- Often the parameters are known (or obtained through separate analysis).
- However: If parameters are unknown, how to carry out combined estimation of x and Θ?
- Liu & West describe a simple approach.

Algorithm

- Liu & West, Combined Parameter and State Estimation: (auxiliary particle filter with state estimation)

Input : Monte Carlo sample $(x_t^{(j)}, \Theta_t^{(j)})$ and weights $w_t^{(j)}$, $j = 1,\dots,N$.

1. For $j = 1,\dots,N$: $\mu_{t+1}^{(j)} = E(x_{t+1} \mid x_t^{(j)}, \Theta_t^{(j)})$
2. Sample an integer $k \in \{1,\dots,\mathrm{N}\}$ with probability $g_{t+1}^{(j)} \propto w_t^{(j)} p(y_{t+1} \mid \mu_{t+1}^{(j)}, m_t^{(j)})$
3. Sample $\Theta_{t+1}^{(k)} \sim N(\cdot \mid m_t^{(k)}, h^2 V_t)$.
4. Sample $x_t^{(k)} \sim p(\cdot \mid x_t^{(k)}, \Theta_{t+1}^{(k)})$.
5. Evaluate weight : $w_{t+1}^{(k)} \propto \dfrac{p(y_{t+1} \mid x_{t+1}^{(k)}, \Theta_{t+1}^{(k)})}{p(y_{t+1} \mid \mu_{t+1}^{(k)}, m_t^{(k)})}$

Repeat 2.-5. until approximation is sufficiently accurate.

$$\mathrm{m}_t^{(j)} = a\Theta_t^{(j)} + (1-a)\overline{\Theta}_t$$

$\overline{\Theta}_t, V_t$: Posterior mean and variance matrix from $\Theta_t^{(j)}$ and weights $w_t^{(j)}$

R Implementation

- To describe the model, the user supplies his own functions as arguments to main SMC function (together with Y and [hyper]parameters).

- R language very suitable for implementation, especially because of
  - Vectorization
  - Built-in statistical functions
  - The possibility of supplying user-defined functions as arguments
  - Ease of visualization and interaction

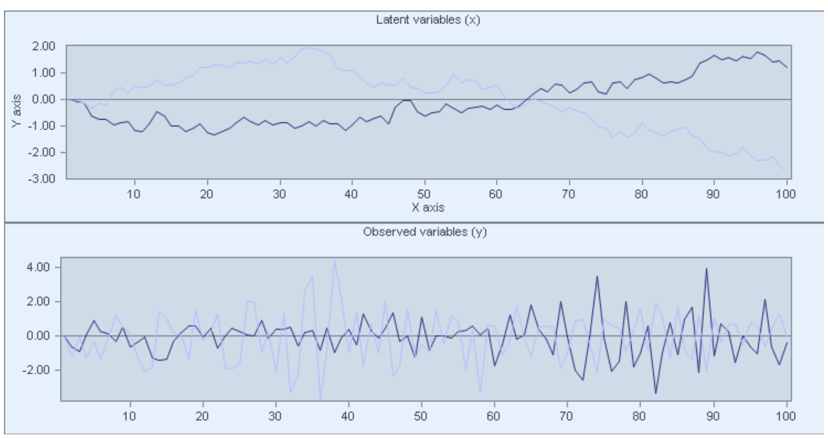
- Quite efficient but still computationally heavy.
  - For large datasets, a C/C++ optimization is needed (we already developed a faster C# version).

Factor Stochastic Volatility Model

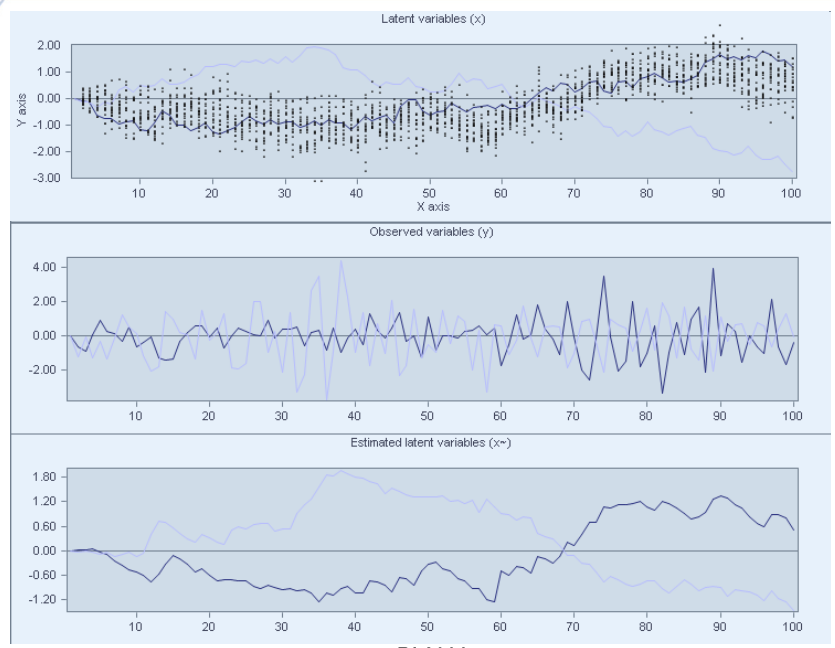
- (similar to the model of Liu & West)

$y_t = \boldsymbol{\alpha}_t + \mathbf{X}\mathbf{f}_t + \boldsymbol{\varepsilon}_t$	Observations
$\boldsymbol{\alpha}_t$	Local series level
$\mathbf{X}$	Factor loadings matrix
$\mathbf{f}_t \sim N(\cdot \mid 0, \mathbf{H}_t)$	Factors
$\mathbf{H}_t = \mathrm{diag}(h_{t1}, \ldots, h_{tk})$	Factor variances
$h_{ti} = \exp(\lambda_{ti})$	
$\boldsymbol{\lambda}_t = \boldsymbol{\mu} + \boldsymbol{\Phi}(\boldsymbol{\lambda}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\gamma}_t$	Log factor variances
$\gamma_t \sim N(\cdot \mid \mathbf{0}, \mathbf{U})$	Innovations
$\mathbf{U}$	Innovations variance matrix
$\boldsymbol{\varepsilon}_t \sim N(\cdot \mid \mathbf{0}, \boldsymbol{\Psi})$	Idiosyncratic noise variances
$\boldsymbol{\Psi} = \mathrm{diag}(\Psi_1, \ldots, \Psi_k)$	

Example, artificial 2-factor model



Example, artificial 2-factor model



Example, FX data

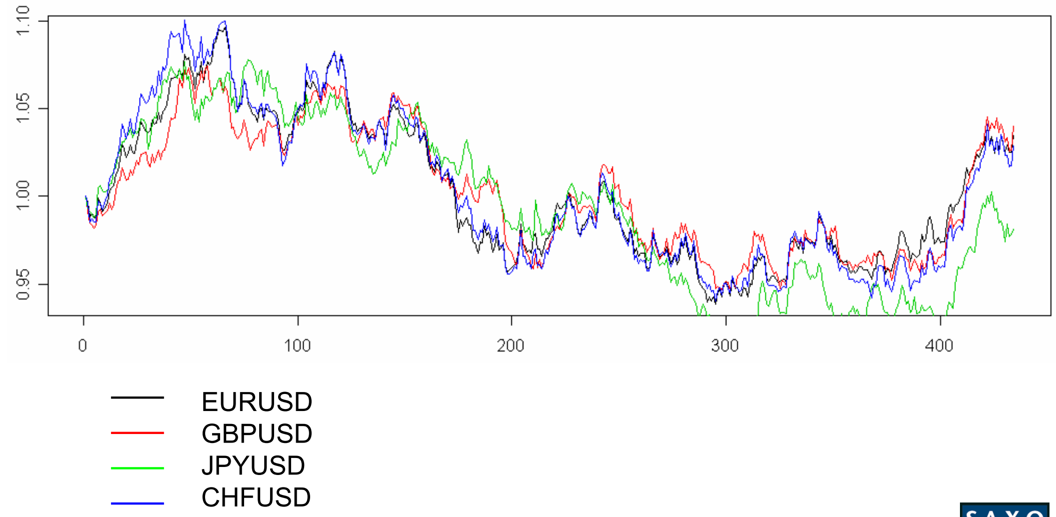
- Model exchange rates with a factor stochastic volatility model.
- Per-minute data
  - EURUSD, GBPUSD, JPYUSD, CHFUSD.
- The log return for currency *i* on day *t* is given by

$$y_{ti} = \log(\frac{s_{ti}}{s_{t-1,i}})$$

where *s* is the spot rate in US dollars.

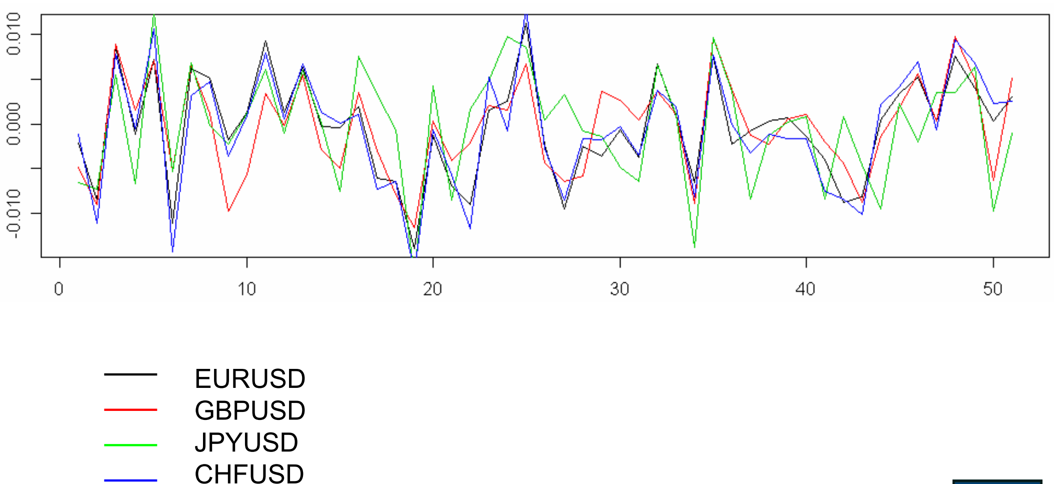
FX data, example

Spot rates. 434 bank days of data. Index 1.0 at 2004-10-01.

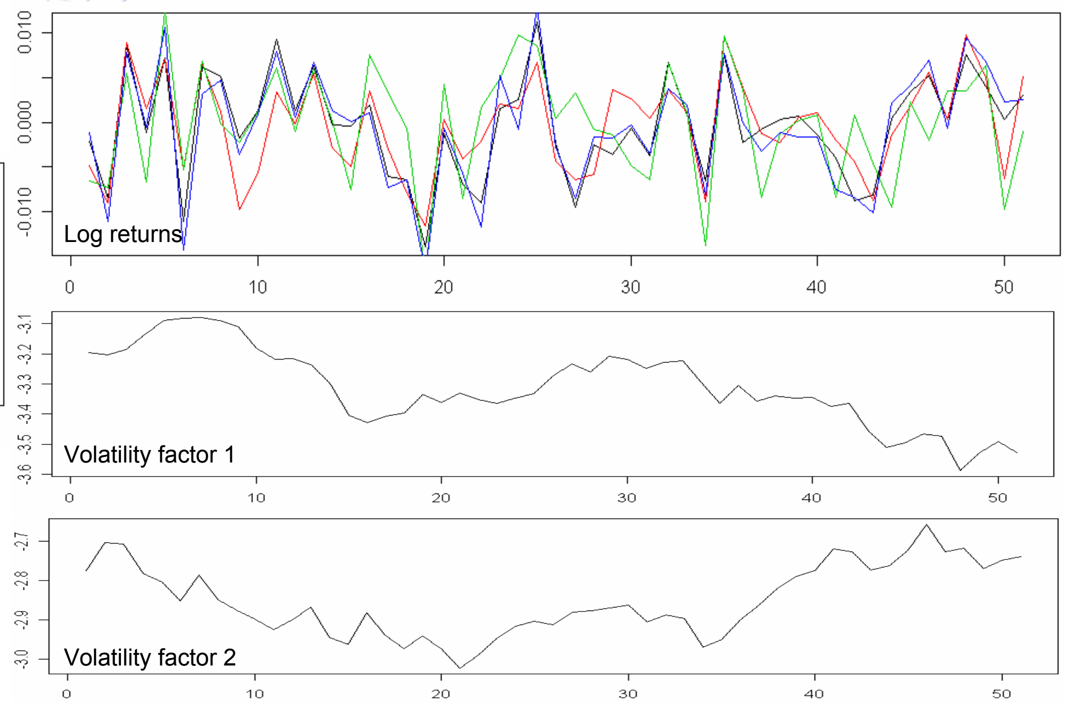


FX data, example

Log return, log (s(t)/s(t-1)) (50 data points)



FX data, results



Conclusion

- R is flexible and powerful enough for implementing efficient particle filters
- For large datasets, however, an optimized C/C++ version is really needed (because of the heavy computational burden).
- Combined parameter and state estimation can be useful but also unstable when there are too many parameters
  - Alternative: Do separate/offline estimation of parameters (using, e.g., full MCMC)
- Package may be forthcoming

SAXO BANK

References

[1] A. Doucet, N. de Freitas and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[2] Liu and West: Combined Parameter and State Estimation in Simulation-Based Filtering, pp. 197-223, in [1].

SAXO BANK

Questions?

- Thanks for your attention.
- Emails:
  » jtl@saxobank.com
  » tj@saxobank.com
- Web site:
  » www.saxobank.com

SAXO BANK