MCMC and Particle Filtering

- Single-move MCMC;
- Block-move MCMC;
- Bootstrap filter;
- Auxiliary Particle Filter;
- APS + parameter estimation

Stochastic Volatility Models
Stochastic volatility models

\[ y_t \sim N(0, \exp(\lambda_t)) \]

\[ \lambda_t = \alpha + \phi \lambda_{t-1} + \omega_t \quad \omega_t \sim N(0, \sigma^2) \]

**Priors**

\[ \lambda_1 \sim N\left(\frac{\alpha}{1 - \phi}, \frac{\sigma^2}{1 - \phi^2}\right) \]

\[ \alpha \sim N(a_\alpha, b_\alpha) \]

\[ \phi \sim TN(a_\phi, b_\phi) \]

\[ \sigma^2 \sim IG(a_\sigma, b_\sigma) \]
Single Move MCMC
(Jacquier et al. 1994)

- Sampling one state at the time:

\[
p(\lambda_t | \lambda_{(-t)}, \Theta) = p(\lambda_t | \lambda_{t-1}, \lambda_{t+1}, \Theta) \\
\propto p(y_t | \lambda_t) p(\lambda_t | \lambda_{t-1}) p(\lambda_{t+1} | \lambda_t)
\]

Density does not have standard form…

… accept/reject step (or possibly MH)

Any other complication?
FFBS (Kim, Shephard, Chib 98)

- Problem: How to filter forward?
  
  - Solution: Approximation through mixtures

\[
\log(y_t^2) = \lambda_t + \log(\nu_t^2)
\]
\[
\lambda_t = \alpha + \phi \lambda_{t-1} + \omega_t
\]
\[
\log(\nu_t^2) \sim \log(\chi^2) \approx \sum_{i=1}^{7} q_i N(a_i, b_i)
\]
FFBS (Kim, Shephard, Chib 98)

- Sample the indicator variable

\[ P(k_t = j | \log(y_t^2), \lambda_t, \Theta) \propto q_j N(\log(y_t^2) | a_j + \lambda_t, b_j) \]

- Forward-Filtering Backward Sampling (as usual).

\[
p(\lambda_1, \ldots, \lambda_T | D_T, \Theta) = p(\lambda_T | D_T, \Theta) \prod_{t=1}^{T-1} p(\lambda_t | \lambda_{t+1}, D_t, \Theta)
\]

- Details in notes from STA214
Particle Filtering

- Observational model

\[ p(y_t | x_t, \Theta) \]

- Markov evolution model

\[ p(x_t | x_{t-1}, \Theta) \]

- Goal: sequentially update posteriors

\[ \cdots \rightarrow p(x_t, \Theta | D_t) \rightarrow p(x_{t+1}, \Theta | D_{t+1}) \rightarrow \cdots \]
Particle Filtering

- Example

\[
y_t = \frac{x_t^2}{20} + \nu_t
\]

\[
x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + \omega_t
\]
Particle Filtering

- **Prior**
  
  $$p(x_t|D_{t-1}, \Theta) = \int p(x_{t-1}|D_{t-1}, \Theta) p(x_t|x_{t-1}, \Theta) dx_{t-1}$$

- **Prediction**
  
  $$p(y_t|D_{t-1}, \Theta) = \int p(y_t|x_t, D_{t-1}, \Theta) p(x_t|D_{t-1}, \Theta) dx_t$$

- **Update**
  
  $$p(x_t|D_t, \Theta) \propto p(x_t|D_{t-1}, \Theta)p(y_t|x_t, D_{t-1}, \Theta)$$
Particle Filtering

- Possible solutions:
  - Extended Kalman-filters
  - Grid-based methods for integration
  - Piecewise linear approximations
  - Sequential importance sampling (particle filters)
Particle Filtering

- Numerical approximations based on “particles” and corresponding weights:

\[
\{x_t^{(j)} : j = 1, \ldots, N\} \quad \{w_t^{(j)} : j = 1, \ldots, N\}
\]

- Prior and posterior can be approximated by the following mixtures:

\[
\hat{p}(x_{t+1}|D_t, \Theta) = \sum_{j=1}^{N} p(x_{t+1}|x_t^{(j)}, \Theta) w_t(j)
\]

\[
\hat{p}(x_{t+1}|D_{t+1}, \Theta) \propto p(y_{t+1}|x_{t+1}, D_t) \sum_{j=1}^{N} p(x_{t+1}|x_t^{(j)}, \Theta) w_t^{(j)}
\]
Bayesian Bootstrap Filter
(Gordon et al. 93)

- At time $t$, suppose we have a set of random samples

$$\{x_t(j) : j = 1, \ldots, N\} \sim p(x_t|D_t, \Theta)$$

- We can evolve the particles through the system to obtain samples from the prior

$$\{x^*_t(j) : j = 1, \ldots, N\} \sim p(x_{t+1}|D_t, \Theta)$$
Bayesian Bootstrap Filter

- Using the prior as a importance density, the set of samples...

\[ \{ x_{t+1}^*(j) : j = 1, \ldots, N \} \]

- …with corresponding weights…

\[ q_j \propto p(y_{t+1}|x_{t+1}^*(j), D_t, \Theta) \]

- …form a weighted sample from the posterior

\[ p(x_{t+1}|D_{t+1}, \Theta) \]
Bayesian Bootstrap Filter

- Why? Sampling Importance Re-sampling (SIR)...

\[
q_j = \frac{p(x^*_t(j)|D_{t+1}, \Theta)}{p(x^*_{t+1}(j)|D_t, \Theta)}
\]

\[
\propto \frac{p(y_{t+1}|x^*_{t+1}(j), D_t, \Theta)p(x^*_{t+1}|D_t, \Theta)}{p(x^*_{t+1}|D_t, \Theta)}
\]

\[
= p(y_{t+1}|x^*_{t+1}(j), D_t, \Theta)
\]

Key cancellation
Bayesian Bootstrap Filter

- Problem: degeneration of the filter
Auxiliary Particle Filter
(Pitt & Shephard 99)

- The idea is to use the mixture approximation to facilitate computations while improving the importance function. The update step will be done by sampling from the following “auxiliary” posterior

\[ p(x_{t+1}, k | D_{t+1}) \propto p(y_{t+1} | x_{t+1}, D_t) p(x_{t+1} | x_t^{(k)}) \]

\[ k = 1, \ldots, N \]

- Drawing from the above joint density and discarding the index k, produce a sample from the approximate posterior density. Again, SIR is used.
**Auxiliary Particle Filter**

- At time $t$, suppose we have a set of random samples and weights
  \[
  \{x_t^{(k)}, w_t^{(k)} : k = 1, \ldots, N\}
  \]

- For each $k$, set the “estimates” and weights
  \[
  \mu_{t+1}^{(k)} = E(x_{t+1} | x_t^{(k)}) \\
  g_{t+1}^{(k)} \propto w_t^{(k)} p(y_{t+1} | \mu_{t+1}^{(k)})
  \]

- Sample the auxiliary variable $j$ with probability given by $g_{t+1}^{(j)}$ followed by
  \[
  x_{t+1}^{(j)} \sim p(x_{t+1} | x_t^{(j)})
  \]
Auxiliary Particle Filter

- Compute the new weights

\[ w_{t+1}^{(j)} \propto \frac{p(y_{t+1} \mid x_{t+1}^{(j)})}{p(y_{t+1} \mid \mu_{t+1}^{(j)})} \]
Back to SVM