Sequential Monte Carlo Estimation of High Dimensional Latent Variable Models

Application to Stochastic Volatility Models

Han Xiao

Nov. 2018

Pennsylvania State Unversity

Motivation

- **Estimate** high-dimensional latent variable models

  Stochastic volatility models: Widely used in practice and option pricing. Application Given asset returns, study the underlying volatility.

Motivation

- **Estimate** high-dimensional latent variable models

  Stochastic volatility models: Widely used in practice and option pricing. Application Given asset returns, study the underlying volatility.

$$y_t = \exp\left(\frac{h_t}{2}\right)\varepsilon_t \tag{1}$$

$$h_{t+1} = \mu + \phi(h_t - \mu) + \sigma\eta_t \tag{2}$$

  where ε and η are iid standard normal distribution.

  - y_t : asset return in reality $\rightarrow$ Observable
  - h_t: risk underlying the asset $\rightarrow$ Latent variable
  - Parameter: $\Theta = \{\mu, \phi, \sigma^2\}$

Motivation

- **Estimate** high-dimensional latent variable models

  Stochastic volatility models: Widely used in practice and option pricing. Application Given asset returns, study the underlying volatility.

$$y_t = \exp\left(\frac{h_t}{2}\right)\varepsilon_t \quad (1)$$

$$h_{t+1} = \mu + \phi(h_t - \mu) + \sigma\eta_t \quad (2)$$

  where ε and η are iid standard normal distribution.

  - y_t : asset return in reality $\rightarrow$ Observable
  - h_t: risk underlying the asset $\rightarrow$ Latent variable
  - Parameter: $\Theta = \{\mu, \phi, \sigma^2\}$

- **Goal**: learn parameters and latent variables recursively
  - use real time data and estimate efficiently

Computational challenge

The difficult part occurs in the latent variable (Equation (2)), i.e., likelihood function is not easily available:

$$f(h_t|y_t,\Theta) \propto \underbrace{f(y_t|h_t,\Theta)}_{observed\ \ return} \underbrace{f(h_t|y_{t-1},\Theta)}_{latent\ \ variable:\ \ volatility} \tag{3}$$

Computational challenge

The difficult part occurs in the latent variable (Equation (2)), i.e., likelihood function is not easily available:

$$f(h_t|y_t,\Theta) \propto \underbrace{f(y_t|h_t,\Theta)}_{observed\ \ return} \quad \underbrace{f(h_t|y_{t-1},\Theta)}_{latent\ \ variable:\ \ volatility} \tag{3}$$

$$f(h_t|y_{t-1},\Theta) = \int \underbrace{f(h_t|h_{t-1},\Theta)}_{proposed\ \ AR(1)} \underbrace{f(h_{t-1}|y_{t-1},\Theta)}_{\color{red}{hard\ \ closed\ \ form}} dh_{t-1} \tag{4}$$

$\Rightarrow$ *Difficult to evaluate the objective function*

Computational challenge

The difficult part occurs in the latent variable (Equation (2)), i.e., likelihood function is not easily available:

$$f(h_t|y_t,\Theta) \propto \underbrace{f(y_t|h_t,\Theta)}_{observed\ \ return} \underbrace{f(h_t|y_{t-1},\Theta)}_{latent\ \ variable:\ \ volatility} \tag{3}$$

$$f(h_t|y_{t-1},\Theta) = \int \underbrace{f(h_t|h_{t-1},\Theta)}_{proposed\ \ AR(1)} \underbrace{f(h_{t-1}|y_{t-1},\Theta)}_{\color{red}{hard\ \ closed\ \ form}} dh_{t-1} \tag{4}$$

$\Rightarrow$ *Difficult to evaluate the objective function*

Challenge 1: Cannot directly draw **sample** from $f(h_t|y_{t-1},\Theta)$

Computational challenge

The difficult part occurs in the latent variable (Equation (2)), i.e., likelihood function is not easily available:

$$f(h_t|y_t,\Theta) \propto \underbrace{f(y_t|h_t,\Theta)}_{observed\ \ return} \underbrace{f(h_t|y_{t-1},\Theta)}_{latent\ \ variable:\ \ volatility} \tag{3}$$

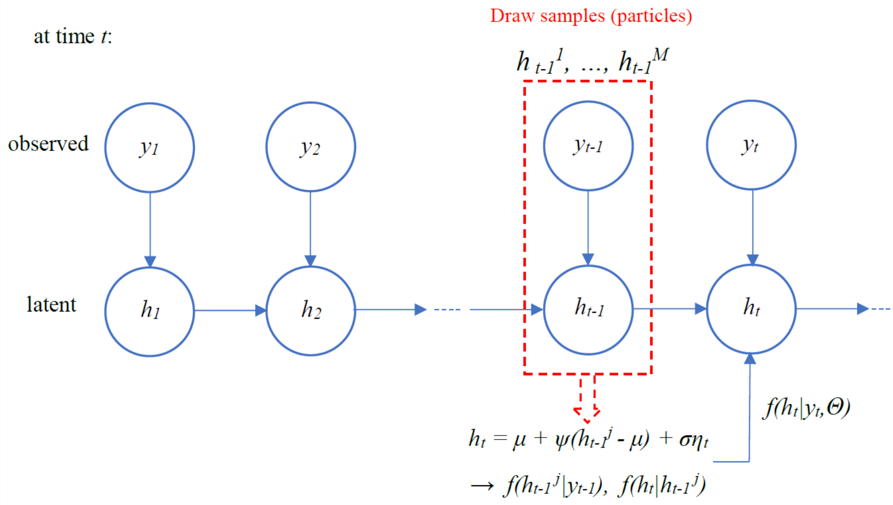
$$f(h_t|y_{t-1},\Theta) = \int \underbrace{f(h_t|h_{t-1},\Theta)}_{proposed\ \ AR(1)} \underbrace{f(h_{t-1}|y_{t-1},\Theta)}_{\color{red}{hard\ \ closed\ \ form}} dh_{t-1} \tag{4}$$

$\Rightarrow$ *Difficult to evaluate the objective function*

Challenge 1: Cannot directly draw **sample** from $f(h_t|y_{t-1},\Theta)$

Challenge 2: Computational **complexity** of simulation increases with the number of time points in data

Intuition for sequential Monte Carlo Method



Sequential Monte Carlo: particle filter algorithm

- Main Idea: draw particles $\{h_{t-1}^j\}$ from filtered distribution $f(h_{t-1}|y_{t-1})$, and derive $f(h_t|y_t)$ using Equation (2). Then sequentially draw particles $\{h_t^j\}$

Sequential Monte Carlo: particle filter algorithm

- Main Idea: draw particles $\{h^j_{t-1}\}$ from filtered distribution $f(h_{t-1}|y_{t-1})$, and derive $f(h_t|y_t)$ using Equation (2). Then sequentially draw particles $\{h^j_t\}$
- Sequentially draw M particles at each time point t $\{h_{t-1}\} = \{h^1_{t-1}, \cdots, h^M_{t-1}\}$ from the filtered distributions: $f(h_{t-1}|y_{t-1})$, where M is the number of particles, then get

$$h_t = \mu + \phi(h^j_{t-1} - \mu)$$

Sequential Monte Carlo: particle filter algorithm

- Main Idea: draw particles $\{h_{t-1}^j\}$ from filtered distribution $f(h_{t-1}|y_{t-1})$, and derive $f(h_t|y_t)$ using Equation (2). Then sequentially draw particles $\{h_t^j\}$
- Sequentially draw M particles at each time point t $\{h_{t-1}\} = \{h_{t-1}^1, \cdots, h_{t-1}^M\}$ from the filtered distributions: $f(h_{t-1}|y_{t-1})$, where M is the number of particles, then get

$$h_t = \mu + \phi(h_{t-1}^j - \mu)$$

This gives Monte Carlo approximations

$$f(h_t|y_{t-1}, \Theta) \approx \frac{1}{M}\sum_{j=1}^{M} f(h_t|h_{t-1}^j, \Theta) \tag{5}$$

$$f(h_t|y_t, \Theta) \propto f(y_t|h_t, \Theta)\frac{1}{M}\sum_{j=1}^{M} f(h_t|h_{t-1}^j, \Theta) \tag{6}$$

Sequential Monte Carlo: particle filter algorithm

- Main Idea: draw particles $\{h_{t-1}^j\}$ from filtered distribution $f(h_{t-1}|y_{t-1})$, and derive $f(h_t|y_t)$ using Equation (2). Then sequentially draw particles $\{h_t^j\}$
- Sequentially draw M particles at each time point t $\{h_{t-1}\} = \{h_{t-1}^1, \cdots, h_{t-1}^M\}$ from the filtered distributions: $f(h_{t-1}|y_{t-1})$, where M is the number of particles, then get

$$h_t = \mu + \phi(h_{t-1}^j - \mu)$$

  This gives Monte Carlo approximations

$$f(h_t|y_{t-1}, \Theta) \approx \frac{1}{M}\sum_{j=1}^{M} f(h_t|h_{t-1}^j, \Theta) \quad (5)$$

$$f(h_t|y_t, \Theta) \propto f(y_t|h_t, \Theta)\frac{1}{M}\sum_{j=1}^{M} f(h_t|h_{t-1}^j, \Theta) \quad (6)$$

- I explore different sampling methods: importance-sampling, bootstrap, auxiliary variable Reference

Methodology: compare with MCMC

MCMC Algorithm

1. Initialize h_1 and $\Theta^{(1)}$
2. Sample h_t from $h_t^{(k+1)}|h_{t-1}^{(k)}, y, \Theta^{(k)}$ for $t = 1, \cdots, n$
3. Sample $\sigma^{2(k+1)}|y, h^{(k+1)}, \phi^{(k)}, \mu^{(k)}$
4. Sample $\phi^{(k+1)}|h^{(k+1)}, \mu^{(k)}, \sigma^{2(k+1)}$
5. Sample $\mu^{(k+1)}|h^{(k+1)}, \phi^{(k+1)}, \sigma^{2(k+1)}$
6. Goto 2

Methodology: compare with MCMC

MCMC Algorithm

1. Initialize h_1 and $\Theta^{(1)}$
2. Sample h_t from $h_t^{(k+1)} | h_{t-1}^{(k)}, y, \Theta^{(k)}$ for $t = 1, \cdots, n$
3. Sample $\sigma^{2(k+1)} | y, h^{(k+1)}, \phi^{(k)}, \mu^{(k)}$
4. Sample $\phi^{(k+1)} | h^{(k+1)}, \mu^{(k)}, \sigma^{2(k+1)}$
5. Sample $\mu^{(k+1)} | h^{(k+1)}, \phi^{(k+1)}, \sigma^{2(k+1)}$
6. Goto 2

Computing is more expensive

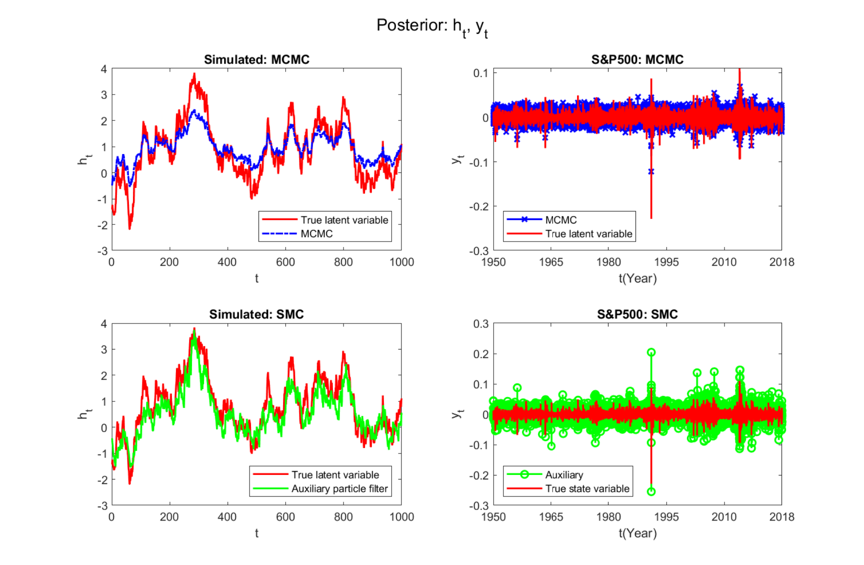
- Need to sample from $h_t^{(k+1)} | h_{t-1}^{(k)}, y, \Theta^{(k)}$ at each time point
- $\{h_t\}$,Θ are highly correlated, resulting in slow convergence ACF

Comparison: Efficiency and Accuracy

- Simulated data: $\mu_0 = 0.5$, $\phi_0 = 0.985$, $\sigma_0^2 = 0.04$
- Real data: Daily S&P 500 returns, 1950 – 2018

	Simulated Data				Real Data			
	MCMC	SISR	Boot	APF	MCMC	SISR	Boot	APF
Computational Efficiency								
ESS	5302	**724**	902	964	1107	**576**	756	826
time	113.106	**0.117**	0.228	0.311	193.733	**2.048**	3.766	4.274
MSE for last latent variable h_N								
mean	**0.013**	0.098	0.100	0.097	1.174	0.134	0.134	**0.132**
std	0.315	**0.041**	0.045	0.043	**0.047**	2.096	2.082	2.003
MSE for parameters								
μ	0.200	0.010	**0.000**	0.090	–	–	–	–
ϕ	0.006	**0.000**	0.000	0.000	–	–	–	–
σ^2	0.004	**0.001**	0.005	0.003	–	–	–	–

Comparison: Latent Variable Estimates



Conclusion

- SMC Strength
  - Efficient: large sample and iteration, more parameters
  - The algorithm can be easily parallelized
  - The computational complexity does not increase with increase in time
  - As the number of particles M increases, the accuracy increases.

Conclusion

- SMC Strength
  - Efficient: large sample and iteration, more parameters
  - The algorithm can be easily parallelized
  - The computational complexity does not increase with increase in time
  - As the number of particles M increases, the accuracy increases.
- SMC Weakness
  - Sensitive to outliers
  - Poor approximation at tails
  - Inference is dominated by a few particles with high weight: Potential solution is auxiliary particle filter

Conclusion

- SMC Strength
  - Efficient: large sample and iteration, more parameters
  - The algorithm can be easily parallelized
  - The computational complexity does not increase with increase in time
  - As the number of particles M increases, the accuracy increases.
- SMC Weakness
  - Sensitive to outliers
  - Poor approximation at tails
  - Inference is dominated by a few particles with high weight: Potential solution is auxiliary particle filter
- Auxiliary particle filter may be better at handling outliers and heavy tails
  - I found some tentative evidence from simulation Density

Thank you

Appendix

- Application
- Auxiliary particle filter algorithm
- MCMC simulation autocorrelation
- Sequence density

Application: Stochastic Volatility Models

1. Option price: Black-Scholes, Heston and Hull-White model
2. Long run risk model
3. Industry

Back

SMC Algorithm: auxiliary particle filter

Back

1. Given $\{h_{t-1}^1, \cdots, h_{t-1}^M\}$ from $f(h_{t-1}|y_{t-1}, \Theta)$ calculate

$$\begin{aligned} \widehat{h}_t^{*j} &= \mu + \phi(h_{t-1}^j - \mu) \\ w_j &= f_N(y_t|exp(\widehat{h}_t^{*j})), \quad j = 1, \cdots, M \end{aligned}$$

   and sample R times with probability $\{w_j\}$. Let the sampled index be $k_1, \cdots, k_R$, and associate these with $\widehat{h}_t^{*k_1}, \cdots, \widehat{h}_t^{*k_R}$

2. For each value of k_j from Step 1 simulate

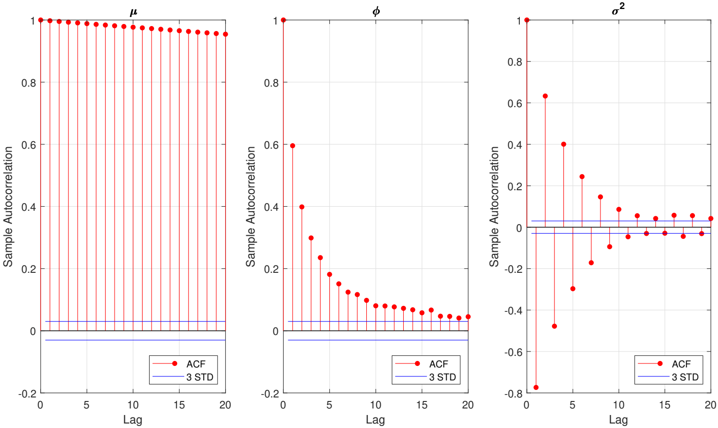
$$h_t^{*j} \sim \mathcal{N}(\mu + \phi(h_{t-1}^{*k_j} - \mu), \sigma^2), \quad j = 1, \cdots, R$$

3. Resample $\{h_t^{*1}, \cdots, h_t^{*R}\}$ with probability

$$\frac{\mathcal{N}(\mu + \phi(h_{t-1}^{*j} - \mu), \sigma^2)}{\mathcal{N}(\mu + \phi(h_{t-1}^{*k_j} - \mu), \sigma^2)}$$

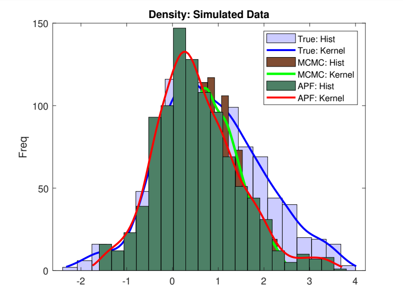
   to produce the filtered sample $\{h_t^1, \cdots, h_t^M\}$ from $f(h_t|y_t, \Theta)$

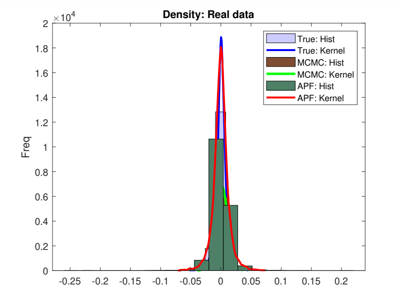
Slow convergence in MCMC: Autocorrelation



Back

Density analysis: simulated and real data





Back

Reference

- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993), Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE Proceedings Part F: Communications, Radar and Signal Processing, 140, 107–113.
- Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. The review of economic studies, 65(3), 361-393.
- Pitt, M. K. and Shephard, N. (1999), Filtering via simulation: auxiliary particle filters, Journal of the American Statistical Association, 94, 590–599.
- Creal, D. (2012). A survey of sequential Monte Carlo methods for economics and finance. Econometric reviews, 31(3), 245-296.

Back