Part IV: Bayesian Inference for DSGE Models

Fei Tan

Chaifetz School of Business
Saint Louis University

Mini-Course Prepared for School of Finance
Zhejiang Gongshang University

June 26, 2018

Introduction

- Why Bayesian paradigm
  - handle sophisticated economic models
  - uncertainty in forecasting & policymaking

- Main references
  - Chib & Greenberg (1995), "*Understanding the Metropolis-Hastings Algorithm*", American Statistician
  - An & Schorfheide (2007), "*Bayesian Analysis of DSGE Models*", Econometric Reviews
  - Herbst & Schorfheide (2015), "*Bayesian Estimation of DSGE Models*", Princeton University Press

The Road Ahead...

- Posterior sampling methods
  - prior, likelihood, and posterior
  - Markov Chain Monte Carlo methods
- DSGE application
  - small new Keynesian DSGE
  - prior distribution
  - tailored randomized block algorithm

Prior, Likelihood, and Posterior

Bayes Theorem

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \propto p(y|\theta)p(\theta)$$

- Learning of random vector $Y = [Y_1, \ldots, Y_n]'$
  - call their realizations $y = [y_1, \ldots, y_n]'$ *data*
  - parametric distribution $\mathbb{P}_\theta$
  - learning of unknown parameter θ
- Bayesian approach treats θ as being random
  - start with *prior* density $p(\theta)$
  - update by *likelihood* function $p(y|\theta)$
  - *posterior* density $p(\theta|y)$ proportional to prior $\times$ likelihood

MCMC Algorithm: Big Picture

A central equation

$$\int_A \int_{\mathbb{R}^d} p(x,y)\pi^*(x)dxdy = \int_A \pi^*(y)dy, \quad \forall A \in \mathcal{B}(\mathbb{R}^d)$$

- What is Markov chain theory doing? Know transition kernel $p(\cdot,\cdot)$, find invariant distribution $\pi^*(\cdot)$

$$\int_A \int_{\mathbb{R}^d} p(x,y)\pi^{(n-1)}(x)dxdy = \int_A \pi^{(n)}(y)dy \to \int_A \pi^*(y)dy$$

- Markov chain Monte Carlo (MCMC) is doing opposite: know $\pi^*(\cdot)$, find corresponding $p(\cdot,\cdot)$ such that

$$\pi^*(x)p(x,y) = \pi^*(y)p(y,x) \quad \text{(reversibility)}$$

- Remark: greatly broaden scope of Bayesian methods though at cost of simulating *dependent* samples

Metropolis-Hastings Algorithm

- Generic MH algorithm
  - initialization: set $\theta^{(0)} = \arg\max p(y|\theta)p(\theta)$
  - recursion: for $k = 1, \ldots, N$

    step 1: draw $\vartheta \sim q(\theta^{(k-1)}, \cdot)$ (proposal density)

    step 2: set $\theta^{(k)} = \vartheta$ with probability of move

    $$\alpha(\theta^{(k-1)}, \vartheta) = \min\left\{ \frac{p(y|\vartheta)p(\vartheta)}{p(y|\theta^{(k-1)})p(\theta^{(k-1)})} \frac{q(\vartheta, \theta^{(k-1)})}{q(\theta^{(k-1)}, \vartheta)}, 1 \right\}$$

    otherwise set $\theta^{(k)} = \theta^{(k-1)}$
  - discard burn-in phase
- MH's choice of $p(\cdot, \cdot)$

$$p_{\mathsf{MH}}(\theta^{(k-1)}, \vartheta) \equiv q(\theta^{(k-1)}, \vartheta)\alpha(\theta^{(k-1)}, \vartheta)$$

satisfies reversibility with invariant distribution $p(\theta|y)$

Random Walk MH

MATLAB pseudo-code

```
function [chain,rej] = RandomWalk_MH(c,Sigma)
[...]
for k = 1:N
    theta = mvt_rnd(chain(k-1,:),c^2*Sigma,inf,1);
    pk_next = PostKer(theta);
    alpha = min([exp(pk_next-pk_last) 1]);
    if rand > alpha                         % reject
        chain(k,:) = chain(k-1,:);
        rej = rej+1;
    else                                    % accept
        chain(k,:) = theta;
        pk_last = pk_next;
    end
end
```

Block-at-a-Time Algorithm

Conditional invariant distributions

$$\int_{A_1}\int_{\mathbb{R}^{d_1}} p_1(x_1, y_1|x_2)\pi^*_{1|2}(x_1|x_2)dx_1dy_1 = \int_{A_1}\pi^*_{1|2}(y_1|x_2)dy_1$$

$$\int_{A_2}\int_{\mathbb{R}^{d_2}} P_2(x_2, y_2|x_1)\pi^*_{2|1}(x_2|x_1)dx_2dy_2 = \int_{A_2}\pi^*_{2|1}(y_2|x_1)dy_2$$

- Product of kernels principle
  - $p_1(x_1, y_1|x_2)p_2(x_2, y_2|y_1)$ has invariant density $\pi^*(x_1, x_2)$
  - underlying Gibbs, MH within Gibbs, & TaRB

Small New Keynesian DSGE

- Dynamic IS relation

$$\hat{y}_t = \mathbb{E}_t \hat{y}_{t+1} + \hat{g}_t - \mathbb{E}_t \hat{g}_{t+1} - \frac{1}{\tau}(\hat{R}_t - \mathbb{E}_t \hat{\pi}_{t+1} - \mathbb{E}_t \hat{z}_{t+1})$$

- New Keynesian Phillips curve

$$\hat{\pi}_t = \beta \mathbb{E}_t \hat{\pi}_{t+1} + \kappa(\hat{y}_t - \hat{g}_t)$$

- Monetary policy rule

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1-\rho_R)\psi_1 \hat{\pi}_t + (1-\rho_R)\psi_2(\hat{y}_t - \hat{g}_t) + \epsilon_{R,t}$$

- Exogenous driving processes

$$\hat{g}_t = \rho_G \hat{g}_{t-1} + \epsilon_{G,t}, \quad \hat{z}_t = \rho_Z \hat{z}_{t-1} + \epsilon_{Z,t}$$

Prior Distribution

Name	Domain	Density	Mean	S.D.
τ	$\mathbb{R}^+$	$\mathbb{G}$	2.00	0.50
κ	$\mathbb{R}^+$	$\mathbb{G}$	0.20	0.10
ψ_1	$(1, \infty)$	$\mathbb{G}$	1.50	0.25
ψ_2	$\mathbb{R}^+$	$\mathbb{G}$	0.50	0.25
$r^{(A)}$	$\mathbb{R}^+$	$\mathbb{G}$	0.50	0.50
$\pi^{(A)}$	$\mathbb{R}^+$	$\mathbb{G}$	7.00	2.00
$\gamma^{(Q)}$	$\mathbb{R}$	$\mathbb{N}$	0.40	0.20
ρ_R	$[0, 1)$	$\mathbb{B}$	0.50	0.20
ρ_G	$[0, 1)$	$\mathbb{B}$	0.80	0.10
ρ_Z	$[0, 1)$	$\mathbb{B}$	0.66	0.15
σ_R	$\mathbb{R}^+$	$\mathbb{IG}$	0.50	0.26
σ_G	$\mathbb{R}^+$	$\mathbb{IG}$	1.25	0.65
σ_Z	$\mathbb{R}^+$	$\mathbb{IG}$	0.63	0.33

Prior Evaluation

MATLAB pseudo-code

```
function logprior = prior_pdf(x,mean,sd,type)

switch type
    case 'G'          % Gamma distribution
        a = mean^2/sd^2;
        b = sd^2/mean;
        logprior = log(gampdf(x,a,b));
    case 'N'          % Normal distribution
        logprior = log(normpdf(x,mean,sd));
    case 'B'          % Beta distribution
        a = -mean*(sd^2+mean^2-mean)/sd^2;
        b = (mean-1)*(sd^2+mean^2-mean)/sd^2;
        logprior = log(betapdf(x,a,b));
    case 'I1'         % Inv-Gamma type-1 distribution
        [...]
end
```

TaRB-MH Algorithm

- A powerful and highly efficient MCMC approach
  - randomize number & components of blocks
  - tailor proposal to posterior location & curvature
- Tailored randomized block (TaRB) algorithm
  - initialization: set $\theta^{(0)} = \arg\max p(y|\theta)p(\theta)$
  - recursion: for $k = 1, \ldots, N$

    step 1: randomize blocks $(\theta_{k,1}, \theta_{k,2}, \ldots, \theta_{k,B_k})$

    step 2: tailor proposal density by optimization routine

    $$q_l(\theta_{k,l}|\theta_{k,-l}, y) = t(\theta_{k,l}|\hat{\theta}_{k,l}, V_{k,l}, \nu)$$

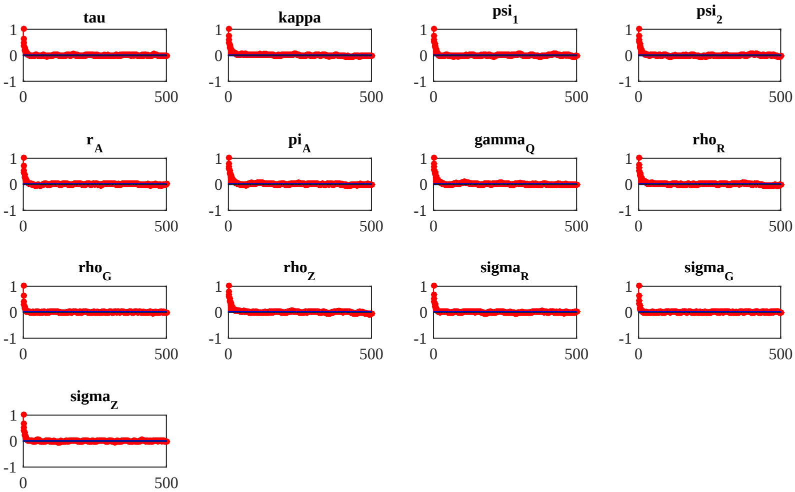
    step 3: update each block with $\alpha_l(\theta_{k,l}, \vartheta_{k,l}|\theta_{k,-l}, y)$
  - discard burn-in phase
- Chib & Ramamurphy (2010), "*TaRB MCMC Methods with Application to DSGE Models*", Journal of Econometrics

Posterior Distribution

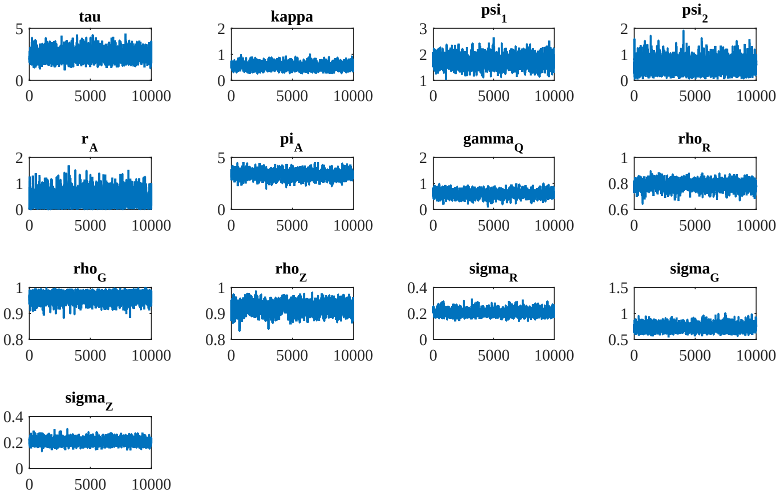
Name	Mean	90% interval	Ineff.
τ	2.435	$[1.712, 3.314]$	8.1
κ	0.543	$[0.362, 0.758]$	16.7
ψ_1	1.738	$[1.400, 2.093]$	10.1
ψ_2	0.570	$[0.198, 1.088]$	13.7
$r^{(A)}$	0.388	$[0.038, 0.869]$	9.4
$\pi^{(A)}$	3.379	$[2.789, 3.968]$	16.6
$\gamma^{(Q)}$	0.605	$[0.399, 0.806]$	16.4
ρ_R	0.791	$[0.735, 0.841]$	17.1
ρ_G	0.963	$[0.933, 0.987]$	7.4
ρ_Z	0.924	$[0.890, 0.956]$	18.0
σ_R	0.208	$[0.173, 0.247]$	10.2
σ_G	0.736	$[0.637, 0.856]$	7.0
σ_Z	0.209	$[0.172, 0.249]$	9.3

NOTES: number of draws = 10,000 after first 1,000 burn-in; computational time = 17m:32s; rejection rate = 45.9%; average number of blocks = 3.4

Autocorrelation Function



Trace Plot


tau
kappa
psi 1
psi 2
r A
pi A
gamma Q
rho R
rho G
rho Z
sigma R
sigma G
sigma Z
0
5000
10000