The Econometrics of DSGE Models

Giuseppe Ragusa
Luiss University

EIEF
Lecture 4: Bayesian computations

April 24, 2015

Outline

- Importance sampling
- Acceptance-rejection sampling
- Monte Carlo Markov Chain
  - Metropolis-Hastings
  - Gibbs sampler
- Readings

Bayesian computations

- To conduct inference using a posterior we need to be able to study its distribution
- Typically we run into these four typical situations:
  - We can simulate from the joint distribution $\theta \sim p(\theta|y)$
  - We cannot simulate from $p(\theta|y)$, but we can simulate from conditionals

$$p(\theta_j|\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_J, y), \quad j = 1, \ldots, J$$

  - We cannot simulate from $p(\theta|y)$ or $p(\theta_j|\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_J, y)$.

* *Simulate* here means that we can obtain draws from the distribution from well established algorithms

Bayesian computations

- We can *simulate* from the joint distribution $\theta \sim p(\theta|y)$
  - Very rare, practically never in real applications — at the best we are able to *evaluate* the joint density
- We cannot simulate from $p(\theta|y)$, but we can simulate from conditionals

$$p(\theta_j|\theta_1,\ldots,\theta_{j-1},\theta_{j+1},\ldots,\theta_J,y), \quad j=1,\ldots,J$$

  - This is a more common situation: we have already see this happening for the linear model with conjugate priors

$$\beta|\sigma^2,y \sim N(\tilde{\beta},s^2A_T^{-1})$$
$$\sigma^2|y\sigma \sim \Gamma^{-1}(\nu_T/2,\nu_T\sigma_T^2/2)$$

- We cannot simulate from $p(\theta|y)$ or $p(\theta_j|\theta_1,\ldots,\theta_{j-1},\theta_{j+1},\ldots,\theta_J,y)$
  - In this case we can only evaluate the joint density up to the proportionality constant, e.g., the likelihood from the linearized DSGE model

$$p(\theta|y) \propto \prod_{t=1}^{T} \phi(y_t;\mu(\theta),\Omega(\theta))p(\theta)$$

Simulation techniques

- We will briefly see the following "simulation methods"
  - Monte Carlo integration
  - Importance Sampling
  - Acceptance-Rejection sampler
  - Monte-Carlo Markov Chain
    - ★ Metropolis-Hastings
    - ★ Gibbs sampler

Monte Carlo integration

- This techniques requires to be able to easily simulate from the *joint* distribution $p(\theta|y)$
- It can calculate *moments* of posteriors, that is,

$$\bar{h} = \int h(\theta)p(\theta|y)d\theta,$$

where $h(\theta)$ is a function which is often called a "*test function*"

  - $h(\theta) = \theta$, $\bar{h}$ is the posterior mean
  - $h(\theta) = \theta^2$, $\bar{h}$ is the uncentered second moment of the posterior
  - $h(\theta) = 1(\theta < u)$, $\bar{h}$ is $\int_{-\infty}^{u} p(\theta|y)d\theta$
  - $h(\theta) = 1(\theta \in A)$, $\bar{h}$ is $\int_A p(\theta|y)d\theta$

Monte Carlo integration, ctd

Main idea

- Sample $\theta_s \sim p(\theta|y)$, then,
  - if $\int |h(\theta)|p(\theta|y)d\theta < \infty$, we have, by the Law of Large numbers

$$\frac{1}{S}\sum_{s=1}^{S} h(\theta_s) \xrightarrow{p} \int h(\theta)p(\theta|y)d\theta$$

  - if $\int h(\theta)^2 p(\theta|y)d\theta < \infty$, we have by the Central Limit Theorem

$$\sqrt{S}\left[\frac{1}{S}\sum_{s=1}^{S} h(\theta_s) - \int h(\theta)p(\theta|y)d\theta\right] \xrightarrow{d} N(0, V_h),$$

$$V_h = E\left[\left(h(\theta) - \int h(\theta)p(\theta|y)d\theta\right)\left(h(\theta) - \int h(\theta)p(\theta|y)d\theta\right)'\right]$$

Monte Carlo integration, ctd

- Very easy to implement, once $p(\theta|y)$ is available
- e.g.,

$$\theta_1, \theta_2|y \sim N(0, I_2)$$

Pseudocode

1: **for** $s \leq S$ **do**
2: Generate θ_s from $p(\theta|y)$
3: **end for**
4: $\bar{h} \leftarrow \frac{1}{S}\sum_{s=1}^{S} h(\theta_s)$

Matlab Code

```
theta_s = randn(S,2)
mean_post = mean(theta_s)
cdf = mean(theta_s<1.96)
set = mean(theta_s<1.96 & ↩
    theta_s>-1.96)
```

- In practice, we never know how to simulate from directly from $p(\theta|y)$.....

Importance Sampling

The objective is the same as before

$$\bar{h} = \int h(\theta) p(\theta|y) d\theta.$$

Suppose that

- we can evaluate the density up to a proportionality constant; that is, given

  $$p(\theta|y) = \gamma(\theta)/Z$$

  we can evaluate $\gamma(\theta)$ but not Z
- e.g., robust linear regression model (Zellner)

$$p(\beta, \sigma^2|y) \propto \underbrace{L(y, \beta, \sigma)}_{\text{student's } t\text{likelihood}} \times \underbrace{N(\beta_0, \sigma^2 A_0^{-1})}_{\text{Normal prior}} \underbrace{\Gamma^{-1}(\nu_0/2, \sigma_0^2/2)}_{\text{Inverted } \Gamma\text{prior}}$$

Importance Sampling

Mean idea:

- We have a distribution $g(\theta)$ from which we can easily simulate
- Since

$$\begin{aligned}\bar{h} &= \int h(\theta)(\gamma(\theta)/Z)d\theta \\ &= \frac{\int h(\theta)\gamma(\theta)d\theta}{Z} \\ &= \frac{1}{Z}\int h(\theta)\frac{\gamma(\theta)}{g(\theta)}g(\theta)d\theta \\ &= \frac{1}{Z}\int w(\theta)\,h(\theta)\,g(\theta)d\theta, \quad w(\theta) = \frac{\gamma(\theta)}{g(\theta)}.\end{aligned}$$

- We approximate

$$\bar{h}_S = \frac{\frac{1}{S}\sum_{s=1}^{S} w(\theta_s)h(\theta_s)}{\frac{1}{S}\sum_{s=1}^{S} w_s},$$

where $\{\theta_1, \ldots \theta_S\}$ is a sample from $g(\theta)$

Importance sampling, ctd.

If $\int |h(\theta)|p(\theta|y)d\theta < \infty$, then

$$\frac{\frac{1}{S}\sum_{s=1}^{S} w(\theta_s)h(\theta_s)}{\frac{1}{S}\sum_{s=1}^{S} w_s} \xrightarrow{p} \int h(\theta)p(\theta|y)d\theta$$

- The choice of the importance density $g(\theta)$ is key in the performance of the sampler
  - the closer is $\gamma(\theta)/Z$ to $g(\theta)$ the better will the algorithm will work
  - if $\gamma(\theta)/g(\theta)$ is unbounded then the variance of $\bar{h}_S$ is infinite: $g(\theta)$ must have thicker tails than $\gamma(\theta)/Z$
- A starting point for $g(\theta)$ is to use the asymptotic approximation of the posterior; thus,

$$\theta_s \sim N(\hat{\theta}^{MLE}, -H(\hat{\theta})^{-1})$$

- Requiring fat tails we can use

$$\theta_s \sim MT(\hat{\theta}, -H(\hat{\theta})^{-1}, \nu)$$

Importance sampling, cts

Algorithm 1 Importance sampling algorithm

1: **for** $s \leq S$ **do**
2: Generate θ_s from g
3: $w_s \leftarrow \gamma(\theta_s)/g(\theta_s)$
4: $w_s \leftarrow w_s / \sum_{s=1}^{S} w_s$
5: **end for**
6: $\bar{h}_S \leftarrow \frac{1}{S} \sum_{s=1}^{S} h(\theta_s) w_s$

Importance sampling

Consider simulating from

$$\theta \sim N(0,1)$$

with importance distribution

$$g(\theta) = \mathscr{T}_3(\theta)$$

```
S = 5000; % Number of simulations
theta_s = trnd(3,S,1); % Draw from importance
w = normpdf(theta_s, 0, 1)./tpdf(theta_s', 3)';
w = w/sum(w);
mean_target = sum(w.*theta_s);
pr_target = sum(w.*(theta_s<1.96 & theta_s>-1.96));
disp('Mean of target density')
disp(mean_target)
disp('Pr(-1.96<theta<1.96) target density')
disp(pr_target)
```

```
Mean of target density                    0.0057
Pr(-1.96<theta<1.96) target density       0.9484
```

Importance sampling

Bad behavior

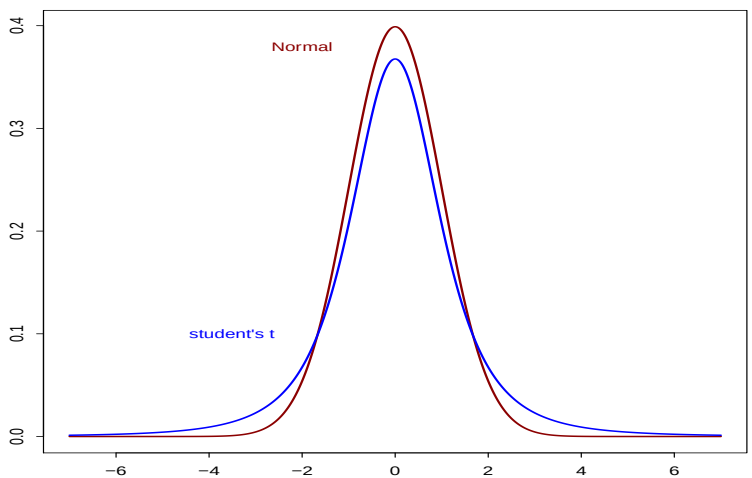
We consider now simulating from

$$\theta \sim \mathscr{T}_3(\theta)$$

with importance distribution

$$g(\theta) = N(0,1)$$

Notice that in this case the *target* has thinner tails than the *importance:*



Importance sampling

Bad behavior

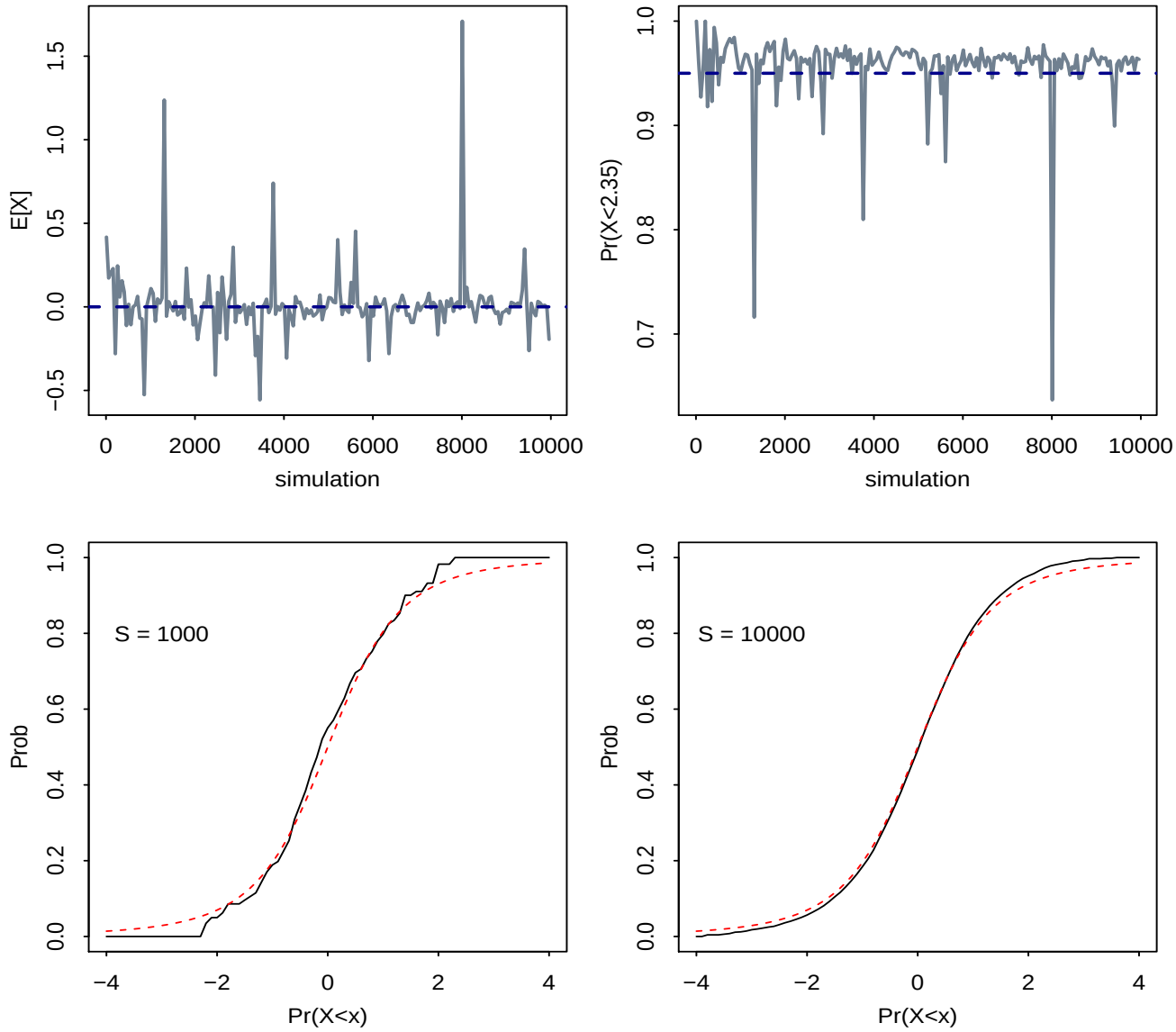


Figure: Performance of the importance sampling algorithm. The target density is

Importance sampling

Application

Consider the robust linear model of Zellner

$$p(\beta,\sigma^2|y) \propto \underbrace{\prod_{t=1}^{T}\left\{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{\nu}{2}\right)\sigma\right\}^{-1}\left[1+\frac{(y-x_t\beta)^2}{\nu\sigma^2}\right]^{-(\nu+1)/2}}_{\text{student's } t\text{likelihood}} \times \underbrace{\sigma^{-2}}_{\text{improper prior}}$$

$$\propto \sigma^{-(T+1)}\prod_{t=1}^{T}\left[1+\frac{(y_t-x_t\beta)^2}{\nu\sigma^2}\right]^{-(\nu+1)/2}$$

$$= \exp\left\{-\frac{(T+1)\log\sigma^2}{2}-\frac{(\nu+1)}{2}\sum_{t=1}^{T}\log\left[1+\frac{(y_t-x_t\beta)^2}{\nu\sigma^2}\right]\right\}$$

We apply the Importance Sampling to this model.

Acceptance/Rejection Algorithm

- Let g be a density that can be simulated by some known method, and suppose there exists a constant c such that $\gamma(\theta) \leqslant c\, g(\theta)$ for all θ. To obtain a sample from $p(\theta|y) = \gamma(\theta)/Z$

Algorithm 2 Acceptance/Rejection sampling

1: **for** $s \leq S$ **do**
2: **repeat**
3: Generate X from g
4: Generate U from $U[0,1]$
5: **until** $U \leqslant \gamma(X)/c\, g(X)$
6: $\theta_s \leftarrow X$
7: **end for**

Acceptance/Rejection sampling

Let $M \equiv \Pr(U \leqslant \gamma(\theta)/cg(\theta))$. Then,

$$\begin{aligned}
\Pr(\theta_s \leqslant y) &= \Pr(Z \leqslant y | U \leqslant \gamma(X)/c\,g(X)) \\
&= \frac{\Pr(X \leqslant y, U \leqslant \gamma(X)/c\,g(X))}{\Pr(U \leqslant \gamma(X)/g(X))} \\
&= \frac{\Pr(X \leqslant y, U \leqslant \gamma(X)/c\,g(X))}{M} \\
&= \frac{\int \Pr(X \leqslant y, U \leqslant \gamma(X)/c\,g(X) | X = x) g(x) dx}{M} \\
&= \frac{\int_{-\infty}^{y} (\gamma(X)/c\,g(X)) g(x) dx}{M} \\
&= \frac{\int_{-\infty}^{y} \gamma(X) dx}{c\,M}.
\end{aligned}$$

As $y \to +\infty$ we have

$$\int_{-\infty}^{y} \gamma(x) dx \to Z,\ \Pr(X \leq y) \to 1.$$

Thus $M = Z/c$, and the proof is complete.

Acceptance/Rejection sampling

The good, the bad, and the ugly

- Key to the algorithm performance is the choice of c
- The acceptance probability is

$$\Pr(U \leqslant \gamma(X)/cg(X)) = \frac{Z}{c},$$

  there is thus a trade-off between acceptance and condition that $\gamma(\theta) \leqslant cg(\theta)$ for all θ
- We would like to set

$$c = \sup_{\theta} \frac{\gamma(\theta)}{g(\theta)}$$

  but this is rarely known
- As the dimensions of the problem get larger, lots of rejections can take place before a useful sample is generated, thus making the algorithm inefficient and impractical.

Monte Carlo Markov Chain

- Method of sampling a target probability distribution by constructing Markov Chain such that the target is the stationary distribution of the chain
- Our task is to find a chain that *converges* to the target distribution
- Large statistical/probabilist literature has traditionally focused on the opposite problem: given a chain, find the stationary distribution
- Markov chain theory is very difficult to master, but the algorithms that results from it are relatively easy to apply

MCMC: Metropolis algorithm

- As before, we let
$$p(\theta|y) = \gamma(\theta)/Z$$
- We need a proposal distribution:
$$q(\theta_t|\theta_{t-1})$$
The notation makes clear that the proposal might depend on the the previous value of θ_t
- We will also assume that $q(\theta_t|\theta_{t-1}) = q(\theta_{t-1}|\theta_t)$, that is the proposal is symmetrical
- Random walk proposal:
$$q(\theta_t|\theta_{t-1}) \propto \exp\left\{-\frac{1}{2}(\theta_t - \theta_{t-1})'(c\Sigma^{-1})(\theta_t - \theta_{t-1})\right\}$$

MCMC: The Metropolis-Hastings

Algorithm 3 Metropolis algorithm

Choose an initial value θ_1.
for $s = 1 : S$ **do**
    Draw θ^* from $q(\cdot|\theta_s)$
    Calculate $r = \min\left\{\frac{\gamma(\theta^*)}{\gamma(\theta_s)}, 1\right\}$
    Draw $U \sim U[0,1]$
    if $U \leqslant r$ **then**
        $\theta_{s+1} = \theta^*$
    else
        $\theta_{s+1} = \theta_s$
    end if
end for

Metropolis algorithm essentials

- We need:
  - θ_1 to initialize the algorithm
  - Σ to calibrate the the proposal distribution
- In theory, these values do not matter....
- ... in practice they do
  - θ_1 is set close to the posterior mode
  - Σ equal to the inverse of the hessian of the log-likelihood
  - c is set in such a way to have an acceptance rate should be between 0.5 and 0.25
  - adaptive versions of the Metropolis algorithm are available

MCMC: The Metropolis algorithm

- To obtain the scale of the proposal density, Σ:
  - Obtain
    $$\hat{\theta}^{MAP} = \arg\max_{\theta \in \Theta} \log \gamma(\theta)$$
    and set
    $$\Sigma = H^{-1} = \left[-\partial \log \gamma(\theta) / \partial \theta \partial \theta' \right]^{-1}$$
- Experiment with different values of c to obtain the desired acceptance rate

MCMC: The Metropolis algorithm

Application

We want to sample from:

$$p(\beta, \sigma^2|y) \propto \underbrace{\prod_{t=1}^{T} \left\{ \Gamma\left(\frac{1}{2}\right) \Gamma\left(\frac{\nu}{2}\right) \sigma \right\}^{-1} \left[1 + \frac{(y - x_t\beta)^2}{\nu\sigma^2} \right]^{-(\nu+1)/2}}_{\text{student's } t\text{likelihood}} \times \underbrace{\sigma^{-2}}_{\text{improper prior}}$$

$$\propto \sigma^{-(T+1)} \prod_{t=1}^{T} \left[1 + \frac{(y_t - x_t\beta)^2}{\nu\sigma^2} \right]^{-(\nu+1)/2}$$

with proposal density:

$$q(\theta_t|\theta_{t-1}) \propto \exp\left\{ -\frac{1}{2}(\theta_t - \theta_{t-1})' H (\theta_t - \theta_{t-1}) \right\}$$

MCMC: The Metropolis algorithm

Application: Robust linear model

```
function [likelihood] = roblinear(theta,y, X, nu)
% Calculate the likelihood of the robust linear model
% Calculate log-kernel
[T k] = size(X);
beta   = theta(1:k)';
sigma2 = exp(theta(k+1)); % Notice parameter transformation
r = y-X*beta;
f = nu*sigma2;
tmp = log(1+r.^2/f);
likelihood = exp(-0.5*(T+1)*log(sigma2) -((nu+1)/2)*sum(tmp));
```

```
function [chain] = metropolis(sim, y, X, nu)
% Metropolis algorithm
[T k] = size(X);
chain = zeros(sim, k+1);
beta_ols = X\y; % equivalent to (X'X)^-1 X'y
s_ols    = ((y-X*beta_ols)'*(y-X*beta_ols)/(T-k))^(1/2);
theta    = [beta_ols', log(s_ols^2)]; % log(sigma2) is sampled
f = @(theta)-roblinear(theta, y, X, nu);
[theta_s,FVAL,EXITFLAG,OUTPUT,GRAD,HESSIAN] = fminunc(f, theta);
Sigma        = inv(HESSIAN);
gamma_s      = roblinear(theta_s, y, X, nu);
for s=1:sim
    % Draw candidate
    theta_star = mvnrnd(theta_s', Sigma);
    % Construct gamma(theta*)
    gamma_star = roblinear(theta_star, y, X, nu);
    % Construct r
    r = min(exp(gamma_star-gamma_s),1);
    % Draw U
    U = rand(1);
    if(U<=r)
        theta_s = theta_star;
    end
    chain(s,:) = theta_s;
end
```

```
% Number of draws
S = 50000;

% Generate "fake" linear model data
% set sedd to replicate
rng(1245);
T = 200;
k = 2;
nu = 1;
y = trnd(nu, T,1);
X = [ones(T,1), randn(T, (k-1))];

% Call metropolis algorithm
chain = metropolis(S, y, X, nu);
```

Robust linear model

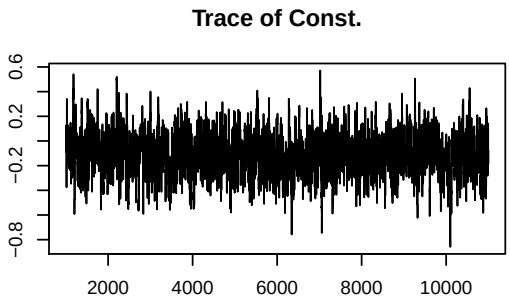
OLS estimates

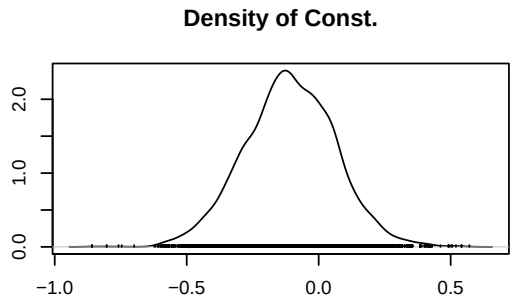
```
## Warning:  package 'MCMCpack' was built under R version 3.1.2
## Warning:  package 'coda' was built under R version 3.1.3
## Warning:  package 'MASS' was built under R version 3.1.2

##
##
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
## The Metropolis acceptance rate was 0.44682
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
##
## Call:
## lm(formula = y ~ X - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -144.609   -0.465    1.001    2.506  128.693
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## Const.   -1.303      1.292  -1.008    0.315
## beta_1    1.175      1.472   0.798    0.426
##
## Residual standard error: 18.14 on 198 degrees of freedom
## Multiple R-squared:  0.007409, Adjusted R-squared:  -0.002617
```

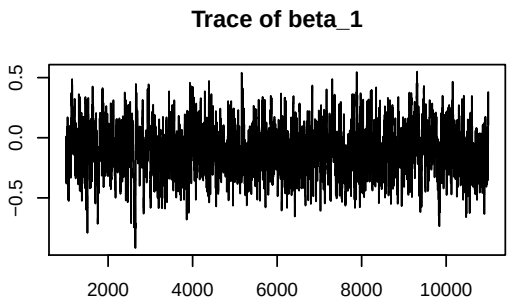
```
##
##
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
## The Metropolis acceptance rate was 0.44682
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean     SD Naive SE Time-series SE
## Const. -0.1100 0.1717 0.001717       0.005848
## beta_1 -0.1011 0.2042 0.002042       0.007027
## sigma2  4.2224 0.6956 0.006956       0.023831
##
## 2. Quantiles for each variable:
##
##           2.5%     25%      50%      75%  97.5%
## Const. -0.4492 -0.2219 -0.11043 0.005888 0.2227
## beta_1 -0.4985 -0.2401 -0.09963 0.035951 0.3033
## sigma2  3.0303  3.7258  4.15981 4.653041 5.7415
```

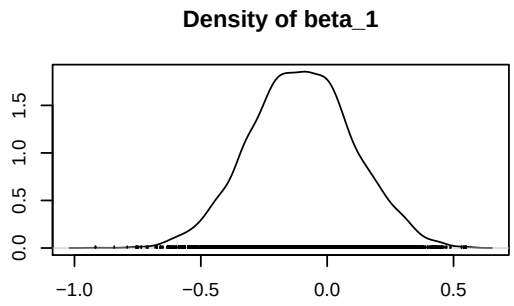
Robust linear model

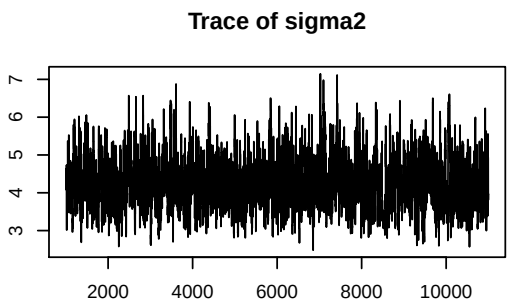
MCMC plot

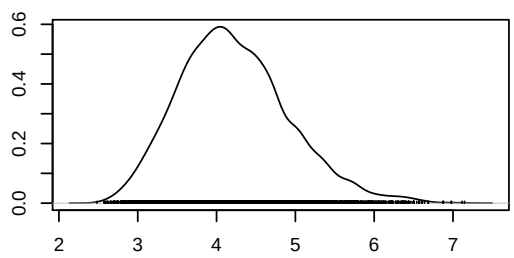












The Metropolis-Hastings algorithm

Algorithm 4 Metropolis-Hastings algorithm

1: Choose an initial value θ_1.
2: **for** $s = 1 : S$ **do**
3: Draw θ^* from $q(\cdot|\theta_s)$
4: Calculate $r = \min\left\{\frac{\gamma(\theta^*)q(\theta_s|\theta^*)}{\gamma(\theta_s)q(\theta^*|\theta_s)}, 1\right\}$
5: Draw $U \sim U[0,1]$
6: **if** $U \leqslant r$ **then**
7: $\theta_{s+1} = \theta^*$
8: **else**
9: $\theta_{s+1} = \theta_s$
10: **end if**
11: **end for**

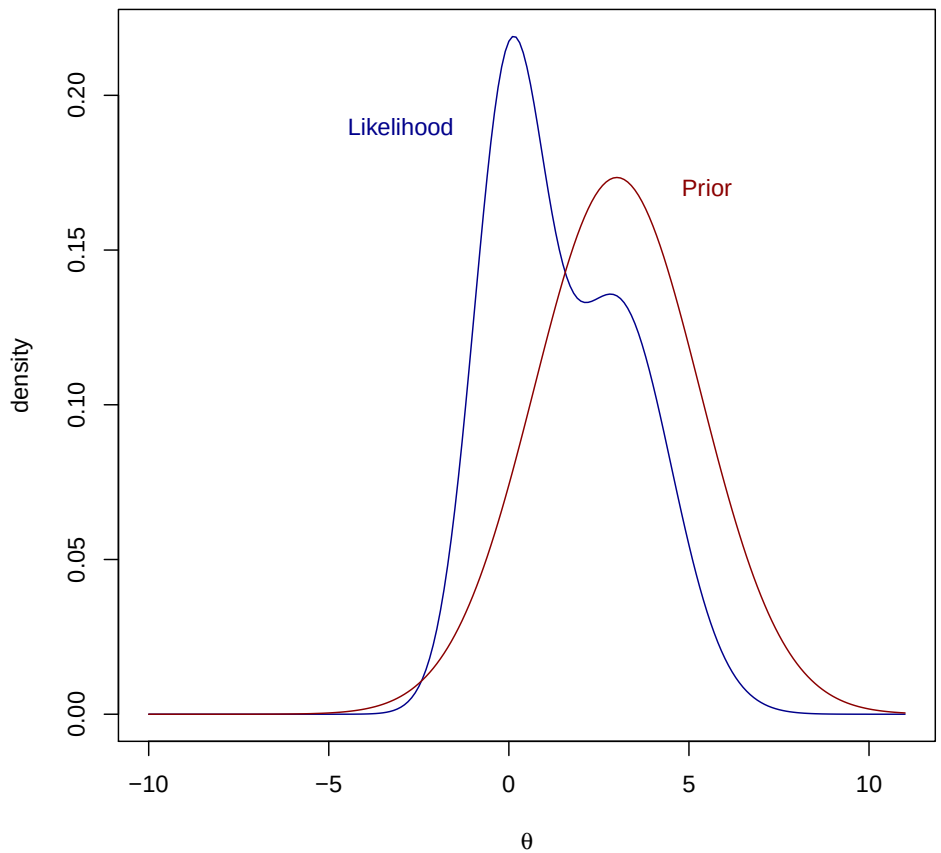
Generalizes the Metropolis algorithm for asymmetric $q(\cdot|\cdot)$.

MCMC: Convergence

- A key issue in the successful implementation of Metropolis-Hastings or any other MCMC sampler is the number of runs (steps) until the chain approaches stationarity (the length of the burn-in period).
- Typically the first 1000 to 5000 elements are thrown out, and then one of the various convergence tests (see below) is used to assess whether stationarity has indeed been reached.
- A poor choice of starting values and/or proposal distribution can greatly increase the required burn-in time, and an area of much current research is whether an optimal starting point and proposal distribution can be found.
- A chain is said to be **poorly mixing** if it says in small regions of the parameter space for long periods of time, as opposed to a **well mixing** chain that seems to happily explore the space.

MCMC: Convergence

A poorly mixing chain can arise because the target distribution is multimodal and our choice of starting values traps us near one of the modes (such multimodal posteriors can arise if we have a strong prior in conflict with the observed data).



MCMC: (Non) Convergence

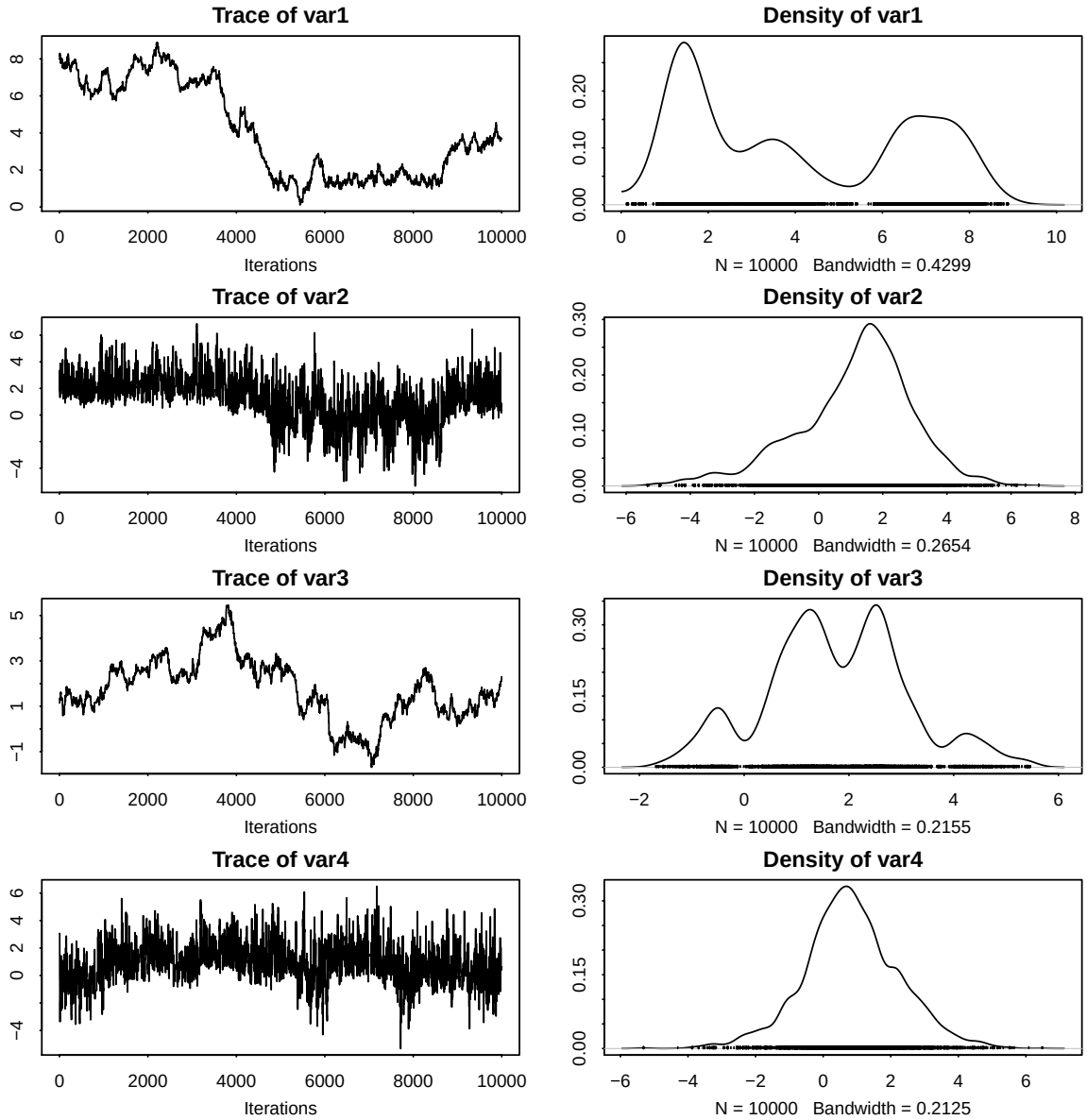


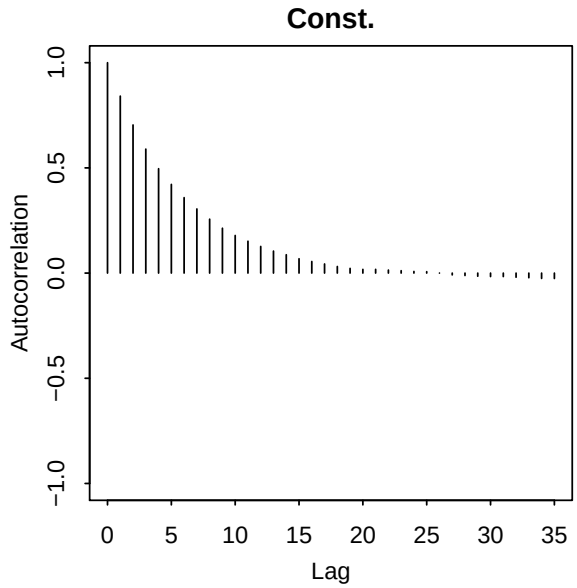
Figure: Example of non-convergent MCMC

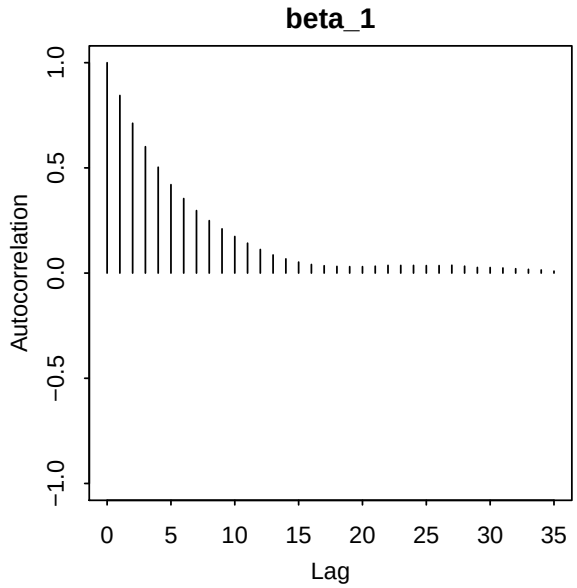
MCMC: Monitoring convergence

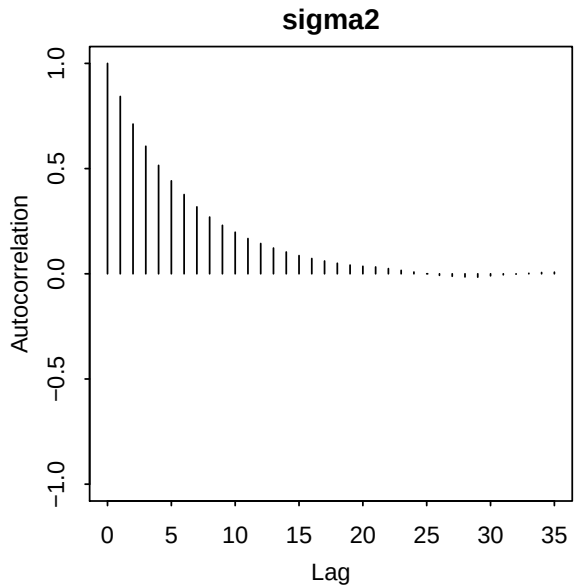
- Convergence is about convergence of distribution
- How to monitor convergence to an invariant distribution?
  - Run multiple chain starting from disperse initial conditions and see whether the chains are "similar" (Gelman and Rubin diagnostic)
  - Trace plots
  - Autocorrelation of draws
  - Geweke's test for the equality of means
  - [Many other ways]

Monitoring convergence

Autocorrelation plot: Robust linear model

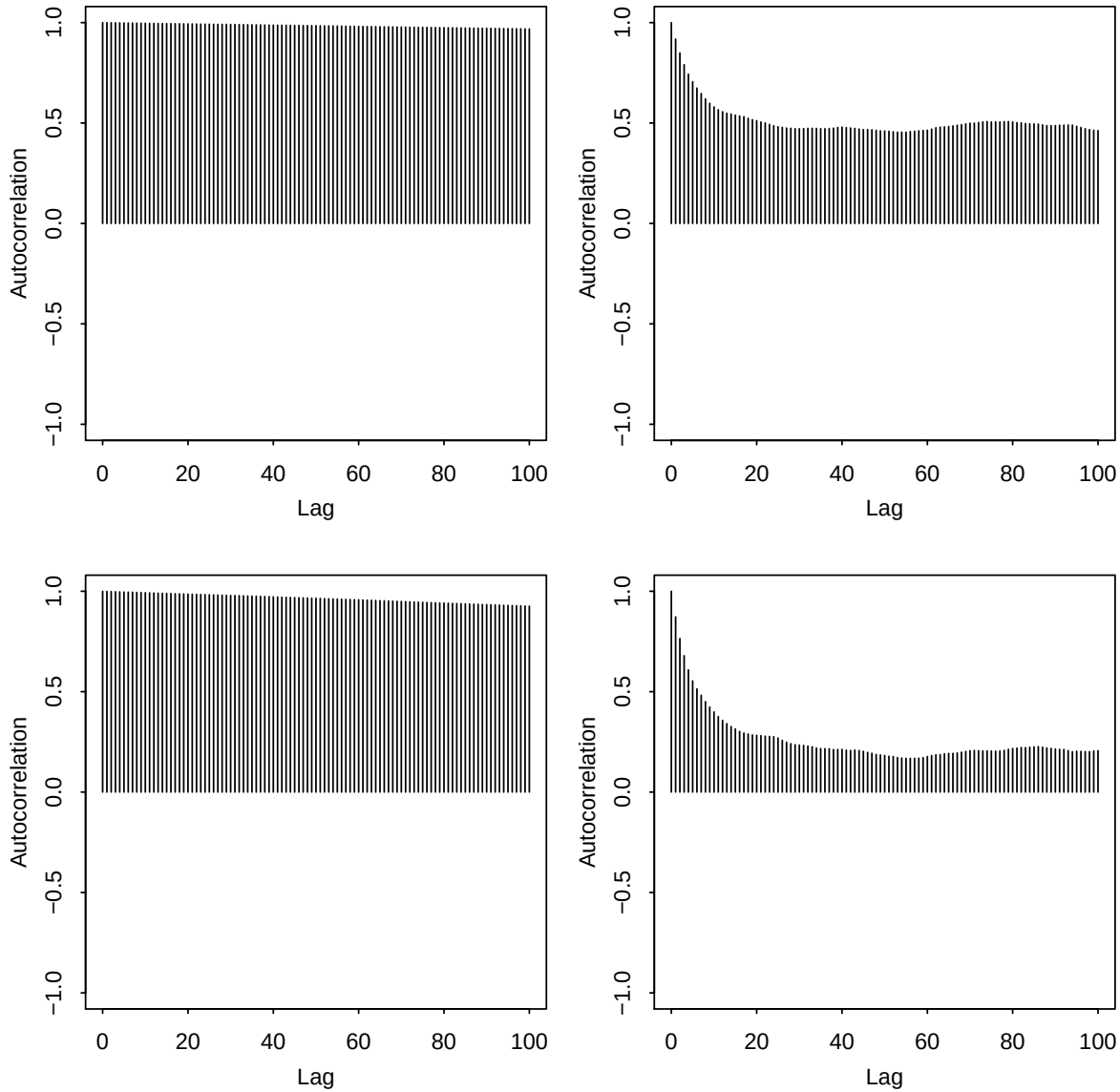






Monitoring convergence

Autocorrelation plot: Of the multimodal posterior



Monitoring convergence

Geweke's diagnostic

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## Const. beta_1 sigma2
##  1.856  1.632  1.245
```

- Geweke's diagnostic for Markov chains is based on a test for equality of the means of the first and last part of a Markov chain (by default the first 10% and the last 50%).
- If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution.
- The test statistic is the difference between the two sample means divided by its estimated standard error.
- The standard error is estimated from the spectral density at zero and so takes into account any autocorrelation.

Gibbs sampler

The problem is to sample from

$$p(\theta_1, \theta_2|y)$$

by conditional distribution

$$p(\theta_1, \theta_2|y) = p(\theta_1|\theta_2, y)p(\theta_2|y)$$

While it is impossible to simulate from $p(\theta_1, \theta_2|y)$, it is feasible to simulate from the conditionals.

- The Gibbs sampler (introduced in the context of image processing by Geman and Geman 1984), is a special case of Metropolis-Hastings sampling wherein the random value is always accepted

Gibbs sampler

Algorithm 5 Gibbs sampler

1: Choose $(\theta_2^{(1)}, \ldots, \theta_k^{(1)})$
2: **for** $s = 1 : S$ **do**
3: Draw $\theta_1^{(s+1)} \sim p(\theta_1 | \theta_2^{(s)}, \theta_3^{(s)}, \ldots, \theta_k^{(s)}, y)$
4: Draw $\theta_2^{(s+1)} \sim p(\theta_2 | \theta_1^{(s+1)}, \theta_3^{(s)}, \ldots, \theta_k^{(s)}, y)$
5: $\vdots$
6: Draw $\theta_k^{(s+1)} \sim p(\theta_k | \theta_1^{(s+1)}, \theta_2^{(s+1)}, \ldots, \theta_{k-1}^{(s+1)}, y)$
7: **end for**

Gibbs sampler

Linear model

The conjugate posteriors for the parameters of the linear model are

$$\beta|y,\sigma^2 \sim N\left(\tilde{\beta}, \sigma^2 A_T^{-1}\right)$$
$$\sigma^2|y \sim \Gamma^{-1}\left(\nu_T/2, \nu_T\sigma_T^2/2\right)$$

Let

$$\begin{aligned}
\hat{\beta} &= (X'X)^{-1}X'Y, \\
s^2 &= (y - X'\hat{\beta})'(y - X'\hat{\beta})/(T-k) \\
A_T &= (A_0 + X'X)^{-1} \\
\tilde{\beta} &= A_T(A_0\beta_0 + X'Y) \\
\nu_T &= \nu_0 + T \\
\nu_T\sigma_T^2 &= \nu_0\sigma_0^2 + (T-k)s^2 + (\beta_0 - \tilde{\beta})A_0(\beta_0 - \tilde{\beta}) + (\hat{\beta} - \tilde{\beta})X'X(\hat{\beta} - \tilde{\beta})
\end{aligned}$$

Gibbs sampler

Linear model

Description of the Gibbs algorithm in this particular case:

1. Draw $\sigma^2_{(1:S)} \sim \Gamma^{-1}\left(\nu_T/2, \nu_T\sigma^2_T/2\right)$
2. For $j = 1 : S$: Draw $\beta_j \sim N\left(\tilde{\beta}, \sigma^2_{(j-1)}A_T^{-1}\right)$

Literature

- Robert, Christian P., and George Casella. Monte Carlo statistical methods. Vol. 319. New York: Springer, 2004.
- Geweke, John, and Michael Keane. “Computationally intensive methods for integration in econometrics.” Handbook of econometrics 5 (2001): 3463-3568.
- Geweke, John. ”Using simulation methods for Bayesian econometric models: inference, development, and communication.” Econometric Reviews 18.1 (1999): 1-73.
- Chib, Siddhartha, and Edward Greenberg. ”Markov chain Monte Carlo simulation methods in econometrics.” Econometric Theory 12 (1996): 409-431.
- Chib, Siddhartha, and Edward Greenberg. ”Understanding the metropolis-hastings algorithm.” The American Statistician 49.4 (1995): 327-335.
- Geweke, John. ”Bayesian inference in econometric models using Monte Carlo integration.” Econometrica: Journal of the Econometric Society (1989): 1317-1339.