Bayesian Methods for DSGE models
Lecture 3
Estimating DSGE models using MCMC methods

Kristoffer Nimark
Cornell University

July 6, 2016

Estimating DSGE models using MCMC methods

The plan:

- The main components in Bayesian estimation
- Sampling from a distribution
- Markov Chain Monte Carlo methods
- The Gibbs Sampler
- The Metropolis-Hastings algorithm

The main components in Bayesian inference

Bayes Rule

Remember Bayes' Rule:

From the symmetry of the joint density

$$p\,(A, B) = p(A \mid B)p(B) = p(B \mid A)p(A)$$

Implying

$$p(A \mid B)p(B) = p(B \mid A)p(A)$$

or

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)}$$

which is THE RULE.

Data and parameters

The purpose of Bayesian analysis is to use the data y to learn about the "parameters" θ

- Parameters of a statistical model
- Or anything not directly observed

The posterior

Replace A and B in Bayes rule with θ and y to get

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}$$

The probability density $p(\theta \mid y)$ then describes what we know about θ, given the data.

The density $p(\theta \mid y)$ is proportional to the prior times the likelihood function

$$\underbrace{p(\theta \mid y)}_{\textit{posterior}} \propto \underbrace{p(y \mid \theta)}_{\textit{likelihood function}} \underbrace{p(\theta)}_{\textit{prior}}$$

The posterior density $p(\theta \mid y)$

The posterior density is often the object of fundamental interest in Bayesian estimation.

- The posterior is the "result".
- We are interested in the entire conditional distribution of the parameters θ

Concepts of Bayesian analysis

Prior densities, likelihood functions, posterior densities are part of all Bayesian analysis.

- Data, choice of prior densities and the likelihood function are the inputs into the analysis
- Posterior densities etc are the outputs

Today we will learn how to simulate the posterior density

Bayesian computation

Posterior simulation

There are only a few cases when the expected value of functions of interest can be derived analytically.

- Instead, we rely on *posterior simulation* and *Monte Carlo integration.*
- Posterior simulation consists of constructing a sample from the posterior distribution $p(\theta \mid y)$

The end product of Bayesian statistics

Most of Bayesian econometrics consists of simulating distributions of parameters using numerical methods.

- A simulated posterior is a numerical approximation to the distribution $p(\theta \mid y)$
- We rely on ergodicity, i.e. that the moments of the constructed sample correspond to the moments of the distribution $p(\theta \mid y)$

The most popular (and general) procedure to simulate the posterior is called the Metropolis-Hastings Algorithm

Posterior Simulation

Various posterior simulators are available and can be used when it is not possible to evaluate the posterior density analytically

The main idea:

- Generate a time series $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, ..., \theta^{S}$ with the property that the sample moments of the time series converges to the moments of the "target density" as S becomes large

The most common simulators are

- The Gibbs Sampler
- Importance Sampling
- Metropolis-Hastings algorithm

Sampling from a distribution

Sampling from a distribution

Example:

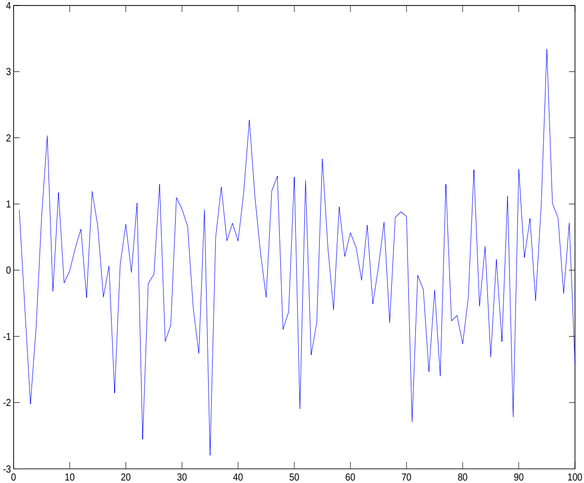
- Standard Normal distribution $N(0,1)$

If distribution is ergodic, sample shares all the properties of the true distribution asymptotically

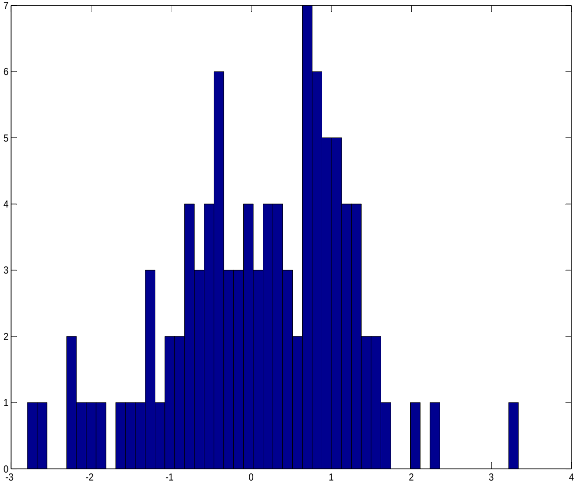
Why don't we just compute the moments directly?

- When we can, we should (as in the Normal distribution's case)
- Not always possible, either because tractability reasons or for computational burden

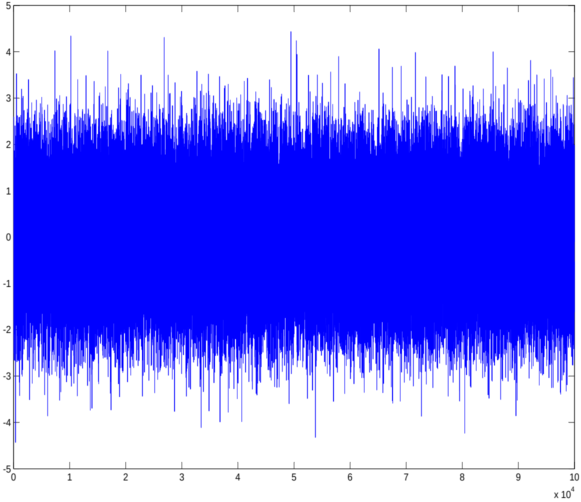
Sample from Standard Normal



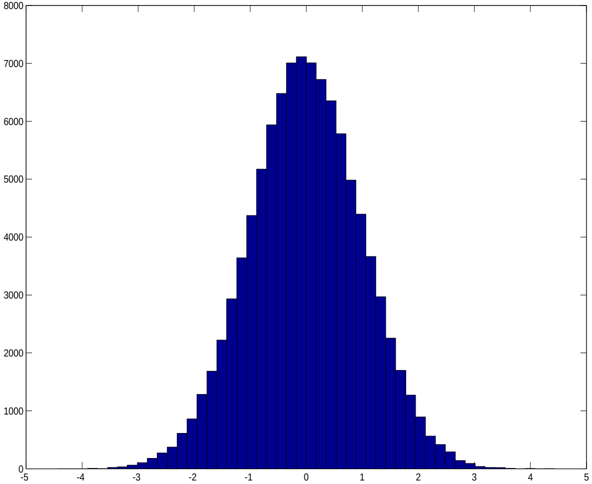
Sample from Standard Normal



Sample from Standard Normal



Sample from Standard Normal



Markov Chains

What is a Markov Chain?

A process for which the distribution of next period variables are independent of the past, once we condition on the current state

- A Markov Chain is a stochastic process with the Markov property
- "Markov chain" is sometimes taken to mean only processes with a countably finite number of states
  - Here, the term will be used in the broader sense

Markov Chain Monte Carlo methods provide a way of generating samples that share the properties of the *target density* (i.e. the object of interest)

Markov Chain Monte Carlo methods

Definition:

A Markov Chain Monte Carlo method for simulating from a distribution f is any method for producing an ergodic Markov chain $\{\theta_s\}_{s=0}^{S}$ whose limiting distribution is f

What is a Markov Chain?

Markov property

$$p(\theta^{(s+1)} \mid \theta^{(s)}) = p(\theta^{(s+1)} \mid \theta^{(s)}, \theta^{(s-1)}, \theta^{(s-2)}, ..., \theta^{(1)})$$

Transition density function $p(\theta^{(s+1)} \mid \theta^{(s)})$ describes the distribution of $\theta^{(s+1)}$ conditional on $\theta^{(s)}$.

- In most applications, we know the conditional transition density and can figure out unconditional properties like $E(\theta)$ and $E\left(\theta^2\right)$
- MCMC methods can be used to do the opposite: Determine a particular conditional transition density such that the unconditional distribution converges to that of the *target distribution*.

Properties of Markov Chains

Irreducibility

- A Markov Chain is said to be *irreducible* if all states communicate

Recurrence

- A Markov Chain is said to be *recurrent* if the probability of returning to each state (at some point in the future) is equal to 1

Irreducibility and recurrence are necessary properties for the chain to be ergodic, i.e. for convergence of sample to the invariant distribution

Not all Markov processes have these properties

The Gibbs Sampler

The Gibbs Sampler

Sometimes conditional distributions of a variable are known and easy to sample from while unconditional distributions are difficult to sample from.

- We can then use the Gibbs Sampler to simulate the unconditional distribution

Consider two parameters of interest $\theta = \left\{\psi^1, \psi^2\right\}$ and that we are interested in sampling from $p(\theta)$ but that it is computationally difficult (or impossible) to do so directly

- However, it may be easy to sample from $p\left(\psi^1 \mid \psi^2\right)$ and $p\left(\psi^2 \mid \psi^1\right)$
- The Gibbs sampler can then be used to simulate a sample from $p(\theta)$.

The Gibbs Sampler for two blocks

1. Start with an arbitrary value ψ_0^1
2. For $s = 0, 1, 2, ..., S$
   2.1 Generate ψ_{s+1}^2 from $p\left(\psi^2 \mid \psi_s^1\right)$
   2.2 Generate ψ_{s+1}^1 from $p\left(\psi^1 \mid \psi_{s+1}^1\right)$
3. Return values $\{\theta_1, \theta_2, \theta_3, ..., \theta_S\}$

The moments of the sample $\{\theta_1, \theta_2, \theta_3, ..., \theta_S\}$ will converge to the moments of $p(\theta)$

The Gibbs Sampler: Example

Multi-variate normal

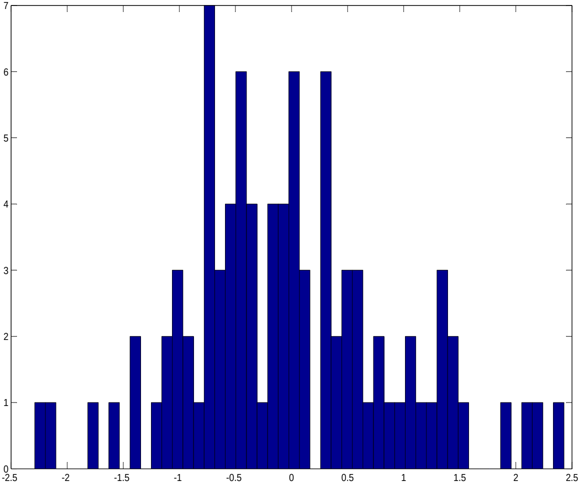
$$\begin{bmatrix} \psi^1 \\ \psi^2 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \right)$$

We can construct an MCMC that simulates the unconditional distribution of ψ^1 and ψ^2 by repeatedly drawing from the conditionals

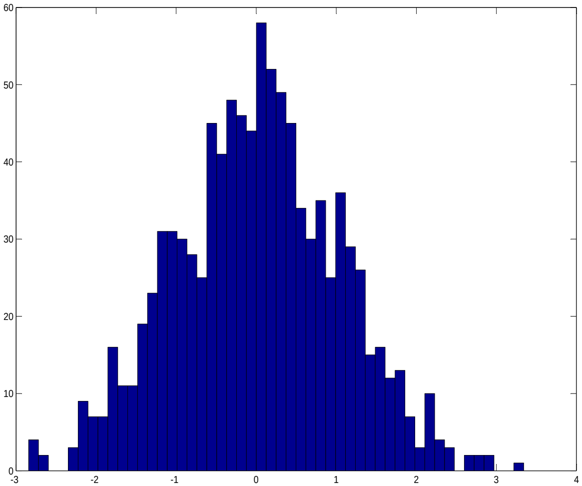
$$\begin{aligned} p\left(\psi^1 \mid \psi^2\right) &\sim N\left(\gamma\psi^2, 1-\gamma^2\right) \\ p\left(\psi^2 \mid \psi^1\right) &\sim N\left(\gamma\psi^1, 1-\gamma^2\right) \end{aligned}$$

Of course, for the multivariate normal, we could draw directly from the unconditional distribution

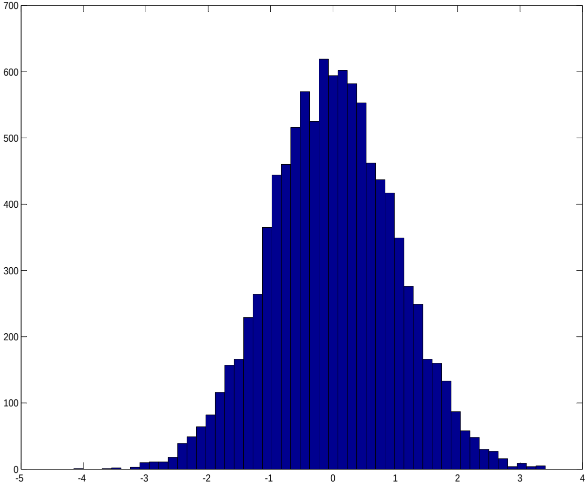
Histogram of ψ^1 100 draws from Gibbs sampler



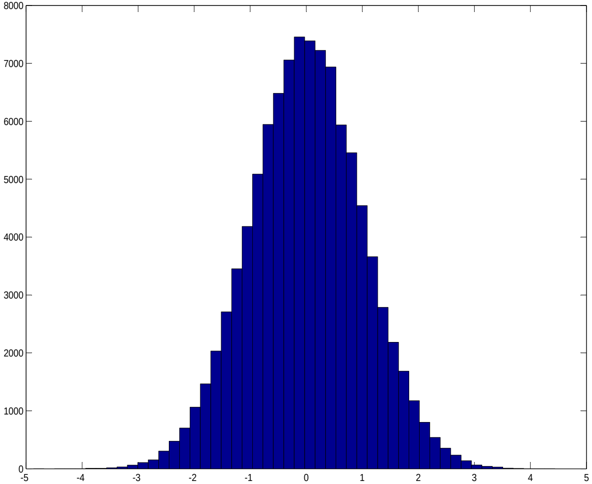
Histogram of ψ^1 1000 draws from Gibbs sampler



Histogram of ψ^1 10 000 draws from Gibbs sampler



Histogram of ψ^1 100 000 draws from Gibbs sampler

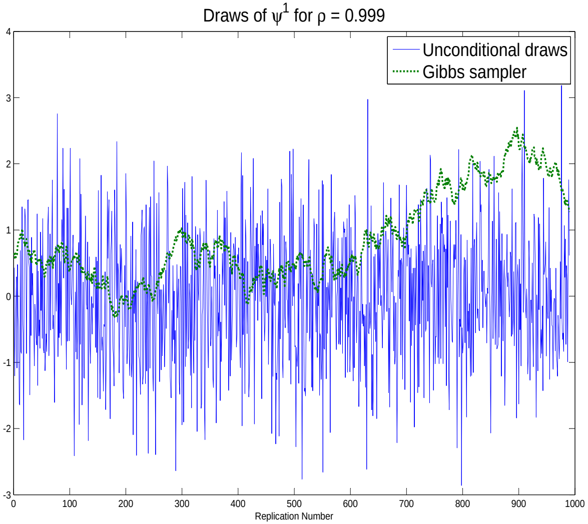


What determines how many draws we need?

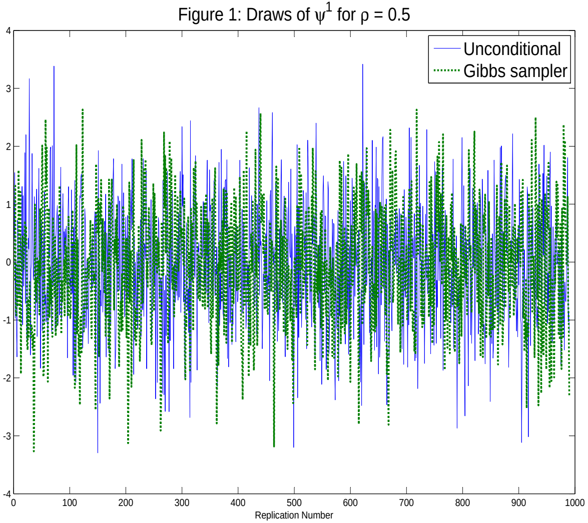
If correlations are strong, i.e. if γ is close to 1 we need more draws.

$$
\begin{aligned}
p\left(\psi^1 \mid \psi^2\right) &\sim N\left(\gamma\psi^2, 1-\gamma^2\right) \\
p\left(\psi^2 \mid \psi^1\right) &\sim N\left(\gamma\psi^1, 1-\gamma^2\right)
\end{aligned}
$$

Sample paths for Gibbs and unconditional draws



Sample paths for Gibbs and unconditional draws



Importance Sampling

Importance sampling

Importance sampling can be used when neither the full posterior nor the conditional posteriors are from a known family of distributions

The importance sampler works well when

1. We can evaluate the posterior density $p(\theta \mid y)$ for a given θ (or a function that is proportional to $p(\theta \mid y)$).
2. We have an easy-to-draw-from distribution $q(\theta)$ that is "close" to $p(\theta \mid y)$

Importance sampling

Let $\theta^{(s)}$ for s=1,2,...,S be a random sample from $q(\theta)$ and define

$$\widehat{g}_S = \frac{\sum_{s=1}^{S} w\left(\theta^{(s)}\right) g\left(\theta^{(s)}\right)}{\sum_{s=1}^{S} w\left(\theta^{(s)}\right)}$$

where

$$w\left(\theta^{(s)}\right) = \frac{p(\theta = \theta^{(s)} \mid y)}{q(\theta = \theta^{(s)})}$$

then $\widehat{g}_S$ converges to $E\left[g\left(\theta\right) \mid y\right]$ as $S \to \infty$

If we are interested in the mean of θ we set $g(\theta) = \theta$ etc.

Metropolis-Hastings Algorithm

Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm is the most general posterior simulator

- Only require that we can evaluate the posterior density
- M-H nests the Gibbs sampler as a special case

Practically all DSGE models are estimated using the M-H Algorithm

Metropolis-Hastings Algorithm

To simulate from the target density $p\left(\theta \mid y\right)$ by the Metropolis-Hastings Algorithm

1. Start with an arbitrary value $\theta^{(0)}$
2. Update from $\theta^{(s-1)}$ to $\theta^{(s)}$ (for $s = 1, 2, ...S$) by
    2.1 Generate a "candidate draw" $\theta^* \sim q(\theta^* \mid \theta^{(s-1)})$
    2.2 Define the acceptance probability

$$\alpha = \min\left(\frac{p\left(\theta^* \mid y\right)}{p\left(\theta^{(s-1)} \mid y\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right) \tag{1}$$

    2.3 Set $\theta^{(s)} = \theta^*$ if $U(0,1) \leq \alpha_s$ and $\theta^{(s)} = \theta^{(s-1)}$ otherwise.
3. Repeat Step 2 S times.
4. Output $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} ..., \theta^{(S)}$

But why does it work?

Simulating distributions using MCMC methods

We can look at a simple discrete state space example:

- θ can take two values $\theta \in \{\theta^L, \theta^H\}$
- Probabilities $p(\theta^L) = 0.4$ and $p(\theta^H) = 0.6$

How can we construct a sequence $\theta_{(1)}, \theta_{(2)}, \theta_{(3)}, ..., \theta_{(J)}$ such that the relative number of occurrences of θ^L and θ^H in the sample correspond to those of the density described by $p(\theta)$?

A two-state Markov Chain

Transition probabilities are defined as

$$\pi_{i,k} = p\left(\theta^{(s+1)} = \theta^{i} \mid \theta^{(s)} = \theta^{k}\right) : i, k \in \{L, H\}$$

Unconditional probabilities solves the equation

$$\begin{bmatrix} \pi_L & \pi_H \end{bmatrix} = \begin{bmatrix} \pi_L & \pi_H \end{bmatrix} \begin{bmatrix} \pi_{LL} & \pi_{HL} \\ \pi_{LH} & \pi_{HH} \end{bmatrix}$$

Problem: Define transition probabilities such that unconditional probabilities equal those of the target distribution

Reverse engineering the transition matrix

We want the chain to spend more time in the more likely state θ^H but not *all* the time.

- Candidate draw $\theta^* = L$ with probability $q(L)$
- Candidate draw $\theta^* = H$ with probability $q(H)$

Finding the right c :

$$\begin{bmatrix} \pi_L & \pi_H \end{bmatrix} = \begin{bmatrix} \pi_L & \pi_H \end{bmatrix} \begin{bmatrix} q(L) & q(H) \\ q(L)\alpha & (1 - q(L)\alpha) \end{bmatrix}$$

Solve the equation $\pi_L = q(L)\pi_l + \alpha q(H)\pi_H$ to get

$$\alpha = \frac{\pi_L q(H)}{\pi_H q(L)}$$

Metropolis-Hastings Algorithm

What is the role of the proposal density?

1. Start with an arbitrary value $\theta^{(0)}$
2. Update from $\theta^{(s-1)}$ to $\theta^{(s)}$ (for $s = 1, 2, ...S$) by
   2.1 Generate a "candidate draw" $\theta^* \sim q(\theta^* \mid \theta^{(s-1)})$
   2.2 Define the acceptance probability

$$\alpha = \min\left(\frac{p\left(\theta^* \mid y\right)}{p\left(\theta^{(s-1)} \mid y\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right) \tag{2}$$

   2.3 Set $\theta^{(s)} = \theta^*$ if $U(0,1) \leq \alpha_s$ and $\theta^{(s)} = \theta^{(s-1)}$ otherwise.
3. Repeat Step 2 S times.
4. Output $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} ..., \theta^{(S)}$

Application: Estimate an SVAR using M-H

Application: Estimate an SVAR using M-H

We want to estimate the structural VAR(1)

$$
\begin{aligned}
\mathbf{y}_t &= A\mathbf{y}_{t-1} + C\mathbf{u}_t : \mathbf{u}_t \sim N(0, I) \\
A &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, C = \begin{bmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{bmatrix}
\end{aligned}
$$

with uniform (improper) priors.

- The parameter vector θ consists of the non-zero elements of A and C.

The Metropolis-Hastings algorithm (again)

1. Start with an arbitrary value $\theta^{(0)}$
2. Update from $\theta^{(s-1)}$ to $\theta^{(s)}$ (for $s = 1, 2, ...S$) by
   2.1 Generate a "candidate draw" $\theta^* \sim q(\theta^* \mid \theta^{(s-1)})$
   2.2 Define the acceptance probability

$$\alpha = \min\left(\frac{p\left(\theta^* \mid y\right)}{p\left(\theta^{(s-1)} \mid y\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right)$$

   2.3 Set $\theta^{(s)} = \theta^*$ if $U(0,1) \leq \alpha_s$ and $\theta^{(s)} = \theta^{(s-1)}$ otherwise.
3. Repeat Step 2 S times.
4. Output $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} ..., \theta^{(S)}$

Application: Estimate an SVAR using M-H

We need:

1. Main code defining starting values and updating the Markov chain for θ
2. A function evaluating the likelihood function $p(y \mid \theta)$ and the prior

Uniform priors

With uninformative priors the acceptance ratio

$$\alpha = \min\left(\frac{p\left(y \mid \theta^*\right) p\left(\theta^*\right)}{p\left(y \mid \theta^{(s-1)}\right) p\left(\theta^{(s-1)}\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right)$$

simplifies to

$$\alpha = \min\left(\frac{p\left(y \mid \theta^*\right)}{p\left(y \mid \theta^{(s-1)}\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right)$$

since

$$\frac{p\left(\theta^*\right)}{p\left(\theta^{(s-1)}\right)} = 1$$

Uniform priors

With uninformative priors the acceptance ratio

$$\alpha = \min\left(\frac{p\left(y \mid \theta^*\right) p\left(\theta^*\right)}{p\left(y \mid \theta^{(s-1)}\right) p\left(\theta^{(s-1)}\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right)$$

simplifies to

$$\alpha = \min\left(\frac{p\left(y \mid \theta^*\right)}{p\left(y \mid \theta^{(s-1)}\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right)$$

since

$$\frac{p\left(\theta^*\right)}{p\left(\theta^{(s-1)}\right)} = 1$$

Symmetric proposal density

With a symmetric proposal density, i.e. if $q(\theta^{(s-1)} \mid \theta^*) = q(\theta^* \mid \theta^{(s-1)})$ the acceptance ratio can be simplified even further

$$\alpha = \min\left(\frac{p\left(y \mid \theta^*\right)}{p\left(y \mid \theta^{(s-1)}\right)}, 1\right)$$

since we then have that

$$\frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})} = 1$$

The likelihood function

We know what the likelihood function for the VAR(1) looks like

$$\begin{aligned} p(y \mid \theta) &= (-Tn/2)\log(2\pi) + \frac{T}{2}\log\left|\Omega^{-1}\right| \\ &- \frac{1}{2}\sum_{t=1}^{T}(y_t - Ay_{t-1})'\Omega^{-1}(y_t - Ay_{t-1}) \end{aligned}$$

and how to evaluate it for a given θ and y.

Choice of proposal density $q(\theta^* \mid \theta^{(s-1)})$

Choice of proposal density $q(\theta^* \mid \theta^{(s-1)})$

Independence Chain Metropolis-Hastings Algorithm

- $q(\theta^* \mid \theta^{(s-1)}) = q(\theta^*)$
- Useful when a good approximation to target density (i.e. the posterior) exists
- Inefficient if $q(\theta^*)$ is very different from target density

Note that $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} \ldots, \theta^{(S)}$ will not be i.i.d.

Choice of proposal density $q(\theta^* \mid \theta^{(s-1)})$

Random-Walk Chain Metropolis-Hastings Algorithm

- $q(\theta^* \mid \theta^{(s-1)}) = N(\theta^{(s-1)}, \Sigma)$ so that

$$\theta^* = \theta^{(s-1)} + z : z \sim N(0, \Sigma)$$

- Normal can be replaced with t-distribution

The RWMH is the most popular choice since it does not requiring much thinking about the proposal density $q(\theta^* \mid \theta^{(s-1)})$

Choice of proposal density $q(\theta^* \mid \theta^{(s-1)})$

Adaptive Random-Walk Chain Metropolis-Hastings Algorithm

- $q(\theta^* \mid \theta^{(s-1)}) = N(\theta^{(s-1)}, \Sigma_{s-1})$ so that

$$\theta^* = \theta^{(s-1)} + z : z \sim N(0, \Sigma_{s-1})$$

- The covariance of the innovation z can be "tuned" as a function of $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} ..., \theta^{(S)}$

One good choice is to set

$$\Sigma_{s-1} = c\Sigma_{\theta s-1}$$

where

$$\Sigma_{\theta s-1} = \frac{1}{s-1} \sum \theta^{(s)} \theta^{(s)\prime}$$

Does any proposal density work?

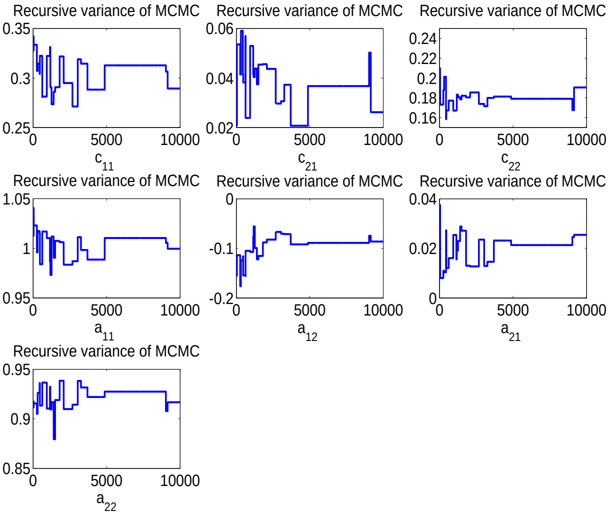
No, but almost:

- Any proposal density with infinite support will work asymptotically, but there is huge variance in terms of how many draws are needed to achieve convergence of the MCMC

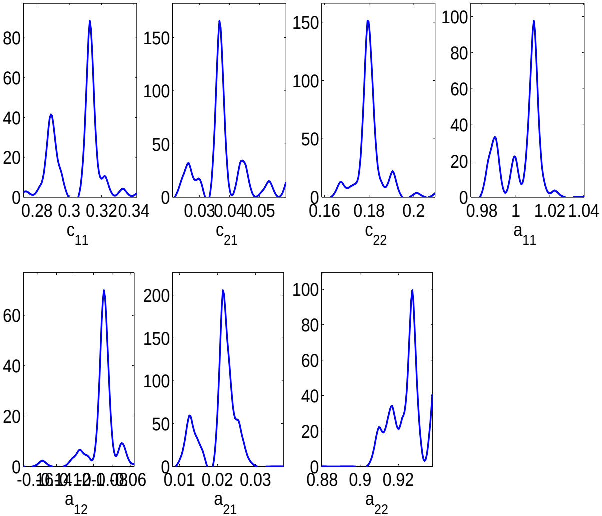
Formally:

$$\bigcup_{x \in supp\ f} supp\ q(\cdot \mid x) \supset supp\ f$$

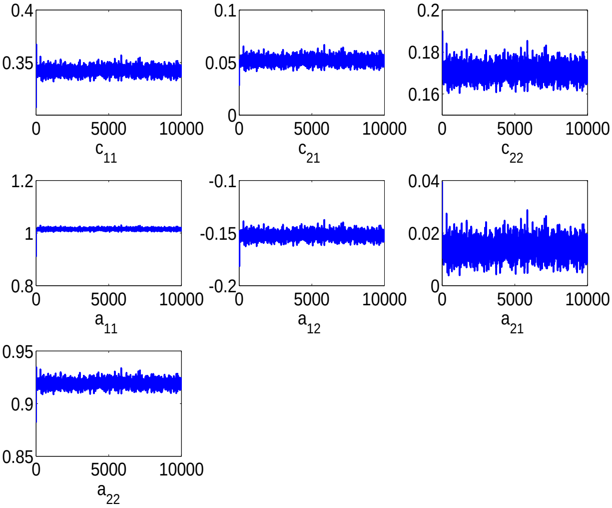
MCMC 10 000 draws non-adaptive



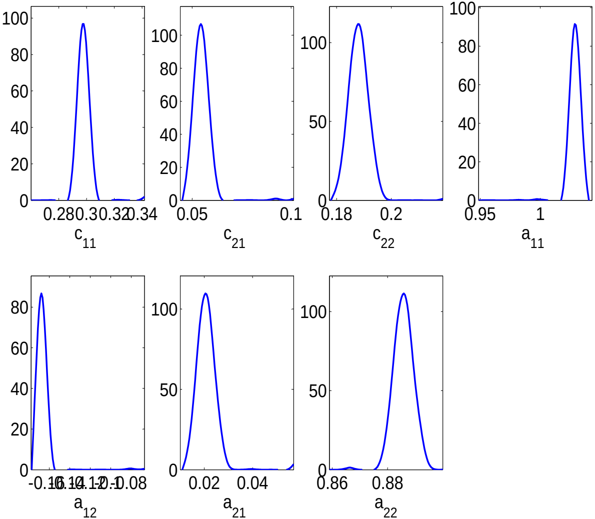
Posterior density 10 000 draws non-adaptive



MCMC 10 000 draws adaptive



Posterior density 10 000 draws adaptive



Tuning the acceptance ratio

For the algorithm to be efficient, not too many nor too few draws should be accepted

- A very high acceptance ratio suggest to small steps so that the chain does not "explore" the parameter space
- A very low acceptance rate suggest that steps are too large, and therefore more likely to be rejected. This also implies that the chain does not "wander".

Acceptance ratio should be in the range $0.2 - 0.3$

Estimating a DSGE model

Estimating a DSGE model

The strategy:

- Specify problems of representative agent and firm and the technologies at their disposal
- Log-linearize model to find approximate first order conditions and budget constraints
- Simulate the posterior density using MCMC methods

The DSGE model as a State Space System

Put the solved model in state space form

$$\begin{aligned} X_t &= AX_{t-1} + Cu_t \\ y_t &= DX_t + v_t \end{aligned}$$

where

$$\begin{aligned} X_t &= x_t, A = \rho, Cu_t = u_t^x \\ y_t &= \begin{bmatrix} \widehat{y}_t \\ \pi_t \end{bmatrix}, D = \begin{bmatrix} -\kappa\frac{\phi_\pi - \rho}{-c} \\ \kappa\gamma\frac{1-\rho}{-c} \\ \phi_\pi\kappa\gamma\frac{1-\rho}{-c} \end{bmatrix} \end{aligned}$$

The DSGE model can be viewed as a function $f(\theta) \rightarrow \{A, C, D, \Sigma_{vv}\}$.

Estimating the model

Estimating the posterior distribution of the parameters

Remember Bayes' Rule:

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}$$

We will use the Metropolis-Hastings algorithm to simulate the posterior $p(\theta \mid y)$

What are the inputs?

- Prior
- Data
- Likelihood function (e.g. the model)

The log likelihood function of a state space system

For a given state space system

$$\begin{array}{rcl} X_t & = & AX_{t-1} + C\mathbf{u}_t \\ \underset{(p\times 1)}{y_t} & = & DX_t + \mathbf{v}_t \end{array}$$

we can evaluate the log likelihood by computing

$$p(y \mid \theta) = -.5 \sum_{t=0}^{T} \left[p \ln(2\pi) + \ln |\Omega_t| + \widetilde{y}_t' \Omega_t^{-1} \widetilde{y}_t \right]$$

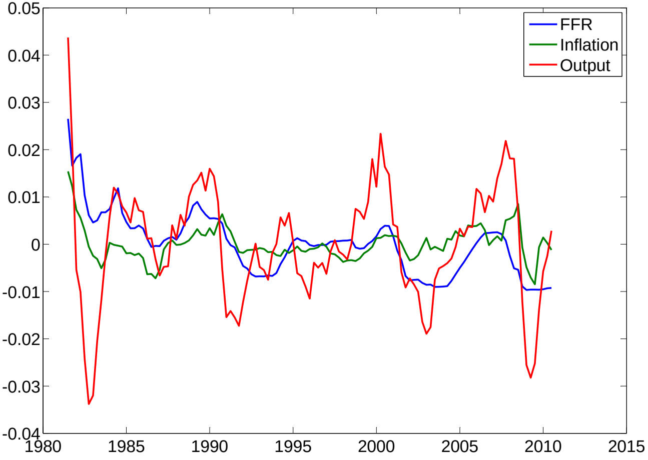
where $\widetilde{y}_t$ are the innovations from the Kalman filter

The data

Which time series to use?

- There are often no perfect, but several imperfect measures, corresponding to the theoretical concepts of the model
- More data is better, but if you have more time series than there shocks in the model, you need to add measurement errors to avoid stochastic singularity

Choice of time series imply a choice for D.



FFR
Inflation
Output
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
1980
1985
1990
1995
2000
2005
2010
2015

Choosing prior

One advantage of DSGE models is that it is often easy to formulate priors about the parameters

- We may have independent evidence on risk aversion, price stickiness, average real interest rates etc
- Priors also help avoid "the dilemma of absurd estimates"

Choosing priors involve:

- Choosing functional forms
- Choosing hyper parameters

Choosing priors

How influential the priors will be for the posterior is a choice:

- It is always possible to choose priors such that a given result is achieved no matter what the sample information is (i.e. dogmatic priors)
- It is also possible to choose priors such that they do not influence the posterior (i.e. so-called non-informative priors)

Important: Priors are a choice and **must** be motivated.

The beta distribution

The beta distribution is a good choice when parameter is in [0,1]

$$P(x) = \frac{(1-x)^{b-1}\, x^{a-1}}{B(a,b)}$$

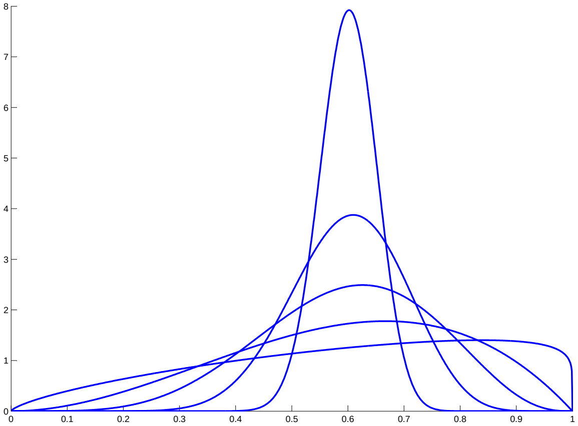
where

$$B(a,b) = \frac{(a-1)!(b-1)!}{(a+b-1)!}$$

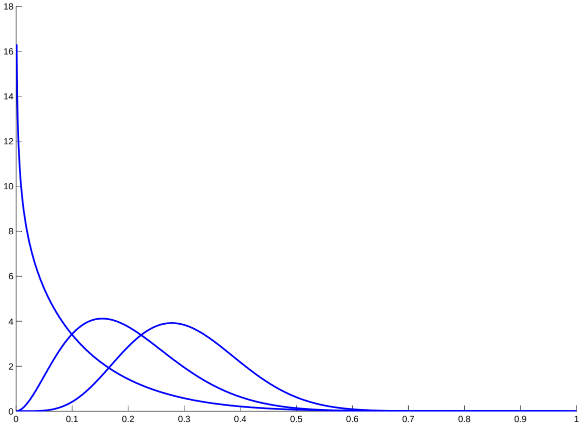
Easier to parameterize using expression for mean, mode and variance:

$$\begin{aligned} \mu &= \frac{a}{a+b}, \quad \widehat{x} = \frac{a-1}{a+b-2} \\ \sigma^2 &= \frac{ab}{(a+b)^2\,(a+b+1)} \end{aligned}$$

Examples of beta distributions holding mean fixed



Examples of beta distributions holding s.d. fixed



The inverse gamma distribution

The inverse gamma distribution is a good choice when parameter is positive

$$P(x) = \frac{b^a}{\Gamma(a)}(1/x)^{a+1}\exp(-b/x)$$

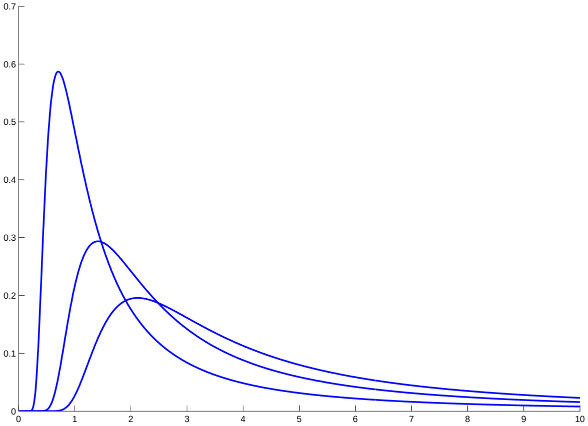
where

$$\Gamma(a) = (a-1)!$$

Again, easier to parameterize using expression for mean, mode and variance:

$$\begin{aligned}
\mu &= \frac{b}{a-1}; a > 1, \quad \widehat{x} = \frac{b}{a+1} \\
\sigma^2 &= \frac{b^2}{(a-1)^2\,(a-2)}; a > 2
\end{aligned}$$

Examples of inverse gamma distributions



Conjugate Priors

Conjugate priors are a particularly convenient:

- Combining distributions that are members of a conjugate family result in a new distribution that is a member of the same family

Useful, but only so far as that the priors are chosen to actually reflect prior beliefs rather than just for analytical convenience

Improper Priors

Improper priors are priors that are not probability density functions in the sense that they do not integrate to 1.

- Can still be used as a form of uninformative priors for some types of analysis
- The uniform "distribution" $U(-\infty, \infty)$ is popular
- Mode of posterior then coincide with MLE.

The prior density $p(\theta)$

Sometimes we know more about the parameters than what the data tells us, i.e. we have some prior information.

The prior density $p(\theta)$ summarizes what we know about a parameters θ before observing the data y.

- For a DSGE model, we may have information about "deep" parameters
  - Range of some parameters may be restricted by theory, e.g. risk aversion should be positive
  - Discount rate is inverse of average real interest rates
  - Price stickiness can be measured by surveys
- We may know something about the mean of a process

Priors on "deep" DSGE parameters

We will use the following informative priors

- $\xi \sim B(0.75, 0.05)$ Price stickiness
- $\gamma \sim N(2, 0.1)$ Risk aversion
- $\beta \sim B(0.99, 0.1)$ Discount factor
- $\phi \sim N(1.5, 0.1)$ Taylor rule coefficient

Improper uniform priors for the remaining parameters

Computing the log of the prior density

In practice, we always compute the log of prior so that with independent priors

$$\begin{aligned} \ln P(\theta) &= \ln p(\theta_1) + \ln p(\theta_2) + \ldots + \ln p(\theta_q) \\ &= \ln p(\gamma) + \ln p(\xi) + \ln p(\beta) + \ln p(\phi) \end{aligned}$$

That is, we can ignore the (constant) probabilities on the uniform priors.

Simulating the posterior

The Random-Walk Metropolis Algorithm

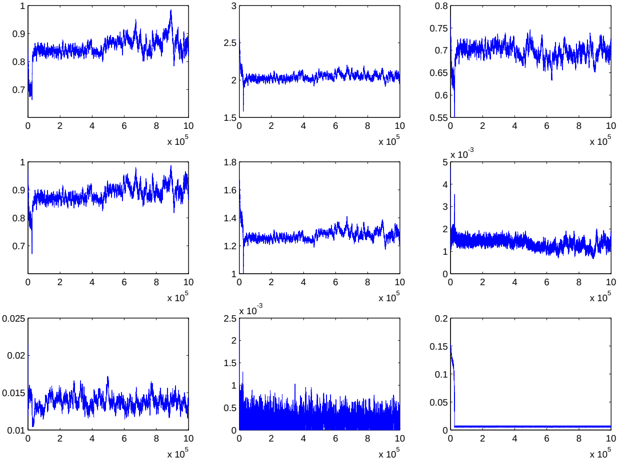
We now have all we need...

1. Start with an arbitrary value $\theta^{(0)}$
2. Update from $\theta^{(s-1)}$ to $\theta^{(s)}$ (for $s = 1, 2, \ldots S$) by
   2.1 Generate a "candidate draw" $\theta^* \sim q(\theta^* \mid \theta^{(s-1)})$
   2.2 Define the acceptance probability

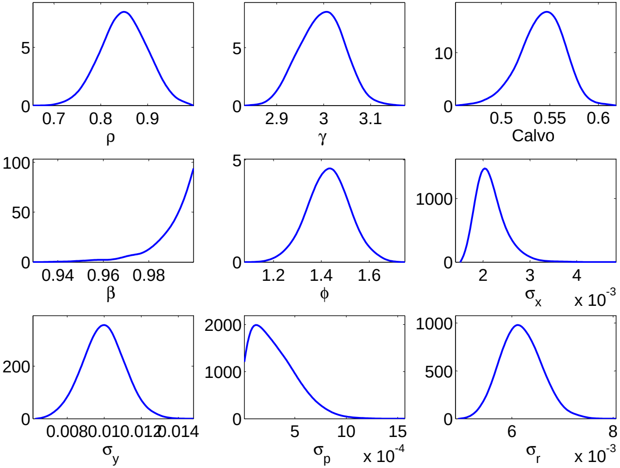
$$\alpha = \min\left(\frac{p\left(\theta^* \mid y\right)}{p\left(\theta^{(s-1)} \mid y\right)} \frac{q(\theta^{(s-1)} \mid \theta^*)}{q(\theta^* \mid \theta^{(s-1)})}, 1\right) \tag{3}$$

   2.3 Set $\theta^{(s)} = \theta^*$ if $U(0,1) \leq \alpha_s$ and $\theta^{(s)} = \theta^{(s-1)}$ otherwise.
3. Repeat Step 2 S times.
4. Output $\theta^{(1)}, \theta^{(2)}, \theta^{(3)} \ldots, \theta^{(S)}$

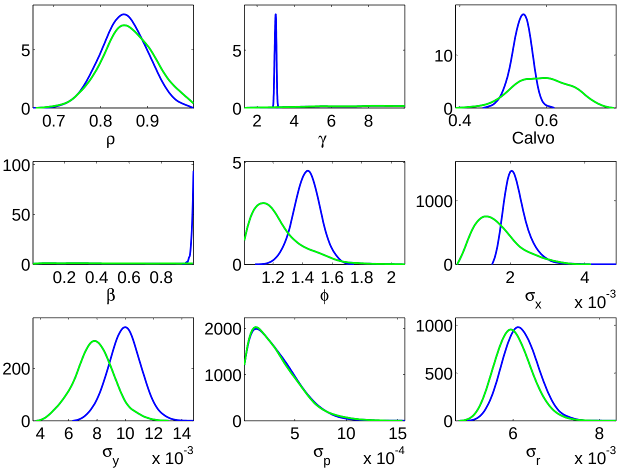
Simulated posterior MCMC



Posterior with informative priors



Informative and non-informative priors



What can we do with the posterior?

- Conduct inference directly on the parameters
- Construct distributions of functions of the posterior
- Construct probabilities of logical statements about functions of parameters and model

That's it for today.