11: Basics of Markov chain Monte Carlo Simulation

Taylor

University of Virginia

Section 1

General Ideas

Introduction

In this section, we discuss **Markov chain Monte Carlo** techniques, which all produce correlated draws from the posterior of interest.

Compared with chapter 12, this chapter mostly seeks to build intuition and mention key ideas.

Overview

Even though we might not even have a time series model, we construct a Markov chain

$$\theta^1, \theta^2, \ldots, \theta^N$$

Even though these draws are correlated, they are all **marginally** distributed according to the posterior

$$p(\theta \mid y),$$

and it is still legitimate to estimate expectations with sample means:

$$E[h(\theta) \mid y] \approx \frac{1}{N} \sum_{i=1}^{N} h(\theta^i).$$

Typical MCMC Output

A 2-parameter model:

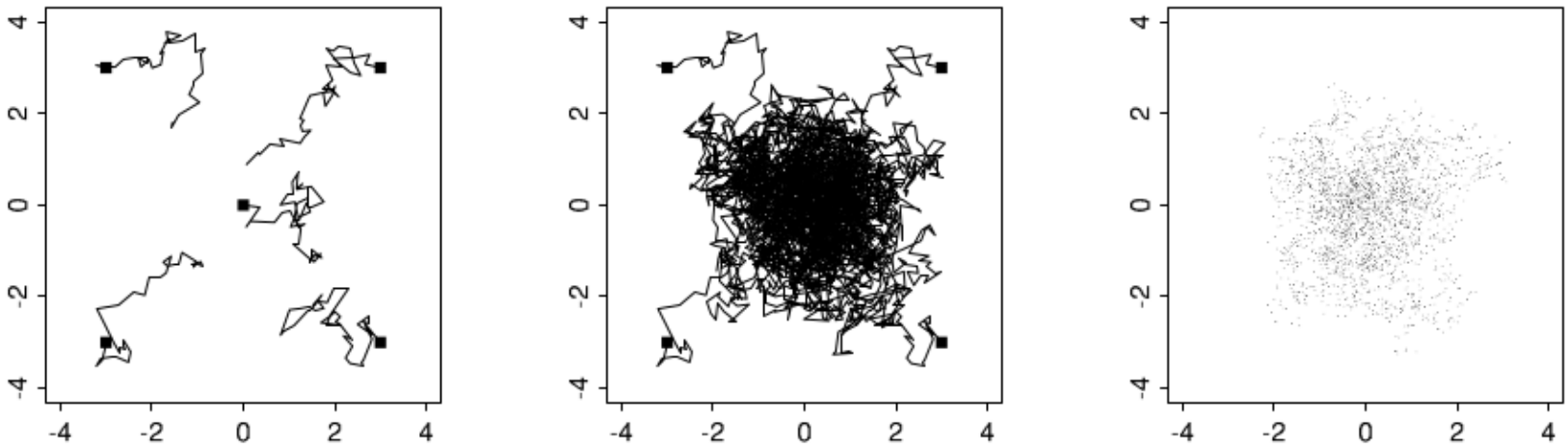


Figure 11.1 *Five independent sequences of a Markov chain simulation for the bivariate unit normal distribution, with overdispersed starting points indicated by solid squares. (a) After 50 iterations, the sequences are still far from convergence. (b) After 1000 iterations, the sequences are nearer to convergence. Figure (c) shows the iterates from the second halves of the sequences; these represent a set of (correlated) draws from the target distribution. The points in Figure (c) have been jittered so that steps in which the random walks stood still are not hidden. The simulation is a Metropolis algorithm described in the example on page 278, with a jumping rule that has purposely been chosen to be inefficient so that the chains will move slowly and their random-walk-like aspect will be apparent.*

Typical MCMC Output

Another 2-parameter model:

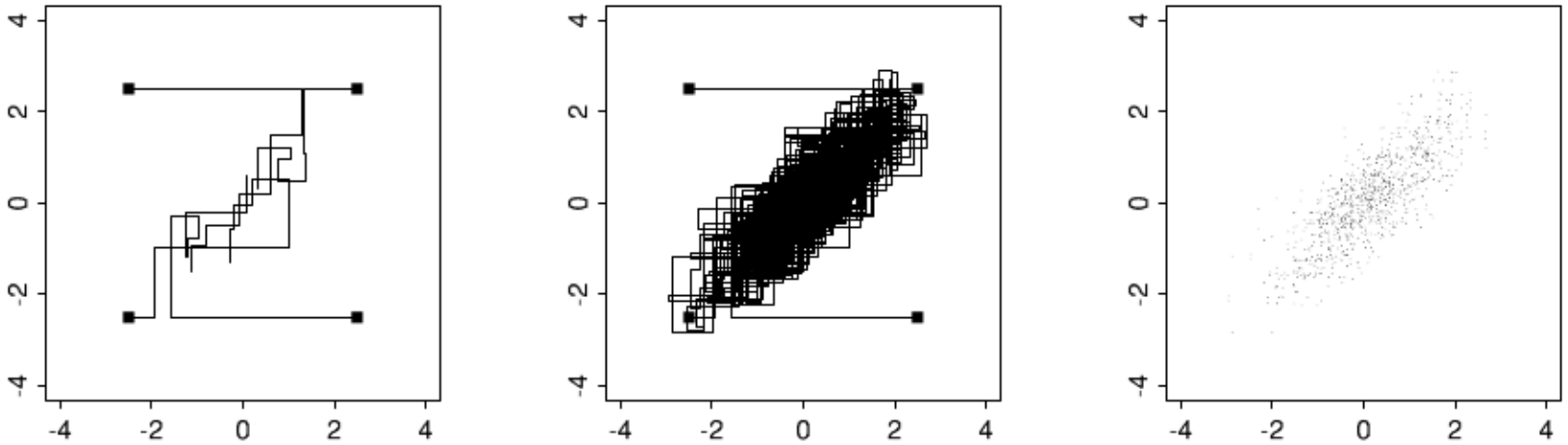
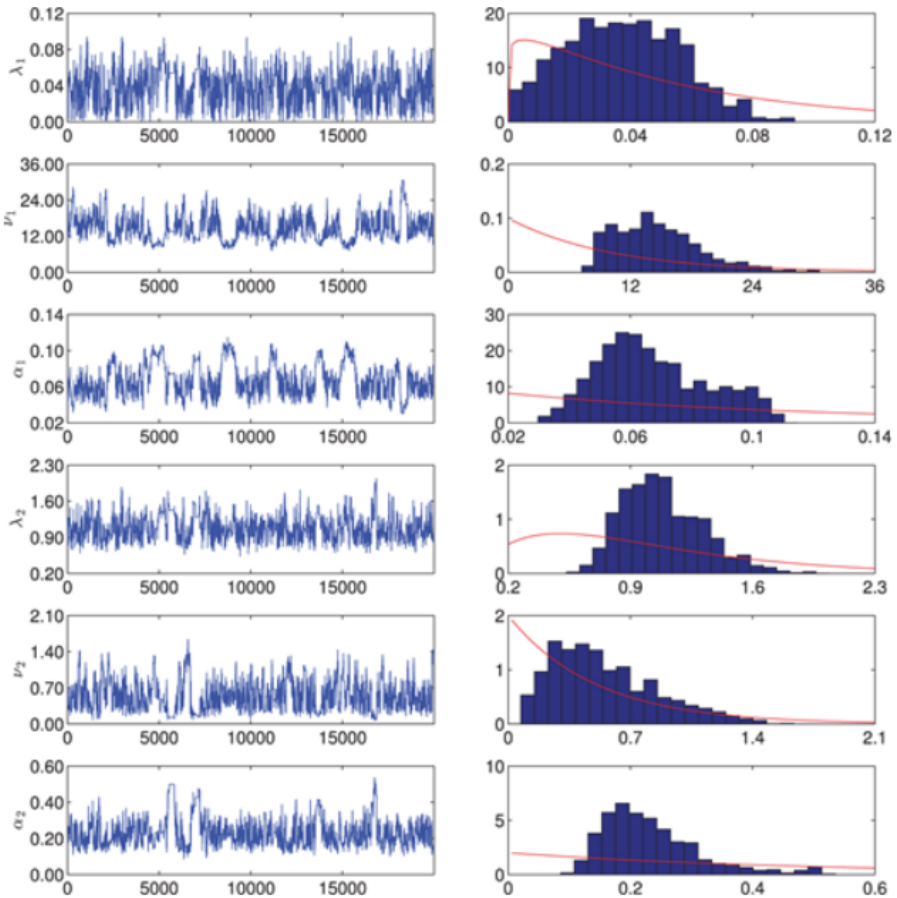


Figure 11.2 *Four independent sequences of the Gibbs sampler for a bivariate normal distribution with correlation $\rho = 0.8$, with overdispersed starting points indicated by solid squares. (a) First 10 iterations, showing the componentwise updating of the Gibbs iterations. (b) After 500 iterations, the sequences have reached approximate convergence. Figure (c) shows the points from the second halves of the sequences, representing a set of correlated draws from the target distribution.*

Typical MCMC Output

A 6-parameter model from:
`https://academic.oup.com/jfec/article/14/2/278/1751519`

Figure 2.



View large Download slide

MCMC output for the two-factor model. Left panels are trace plots of parameters. In the right

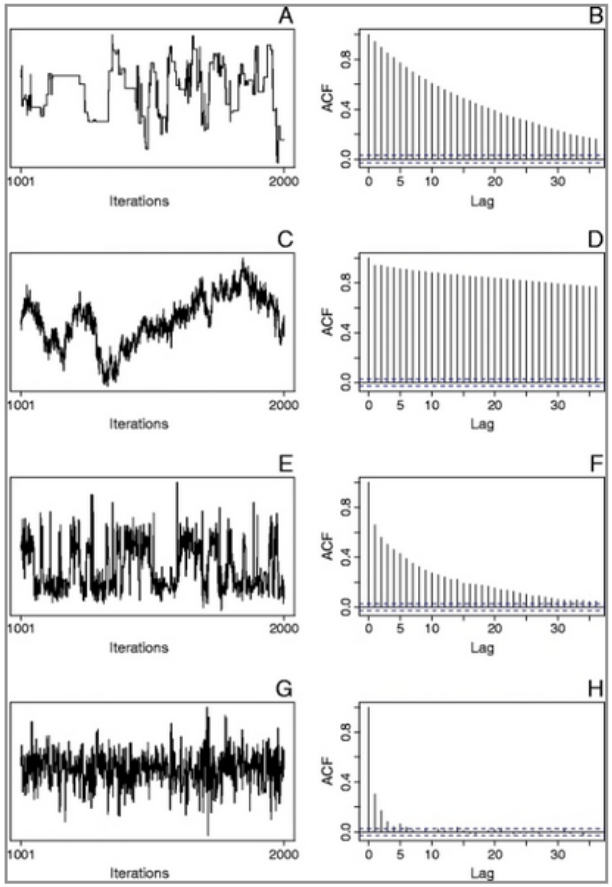
Variance of our Estimator

Let $\sigma^2 = \text{Var}[h(\theta^i)]$, $\rho(h) = \text{Corr}\left(h(\theta^i), h(\theta^{h+i})\right)$. If all of these variances are finite, then

$$
\begin{aligned}
N \,\text{Var}\left[\frac{1}{N}\sum_{i=1}^{N} h(\theta^i)\right] &= N \,\text{Cov}\left(\frac{1}{N}\sum_{i=1}^{N} h(\theta^i), \frac{1}{N}\sum_{j=1}^{N} h(\theta^j)\right) && \text{(defn.)} \\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} \text{Cov}\left(h(\theta^i), h(\theta^i)\right) && \text{(bilinearity)} \\
&= \sigma^2\left\{1 + 2\sum_{h=1}^{N}\frac{N-h}{N}\rho(h)\right\} && \text{(count diagonally)} \\
&\to \sigma^2 \underbrace{\left\{1 + 2\sum_{h=1}^{\infty}\rho(h)\right\}}_{\text{correlation is bad!}}
\end{aligned}
$$

Typical MCMC Output

Assessing the integrated autocorrelation with acf plots:



bad, bad, less bad, good.

`https://openi.nlm.nih.gov/detailedresult?img=PMC3218285_13428_2011_114_Fig10_HTML&req=4`

Another issue: has there been convergence?

The previous expression assumes each draw is distributed according to the posterior.

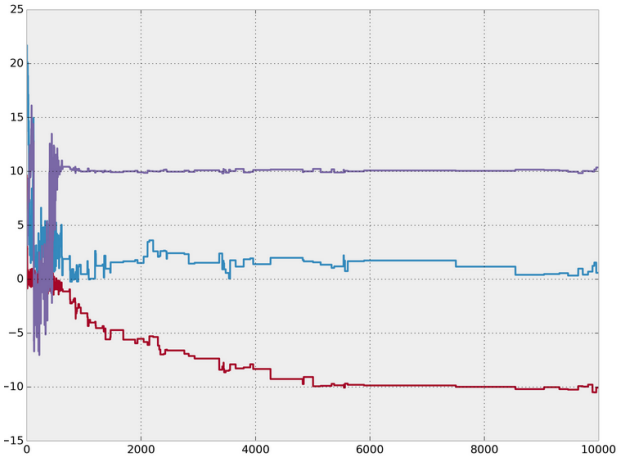
If we start the chain far away from the mode, how long until it converges? How can we be sure it has converged? If you don't run your algorithm for long enough, your answer will be very wrong.

Another issue: has there been convergence?

The previous expression assumes each draw is distributed according to the posterior.

If we start the chain far away from the mode, how long until it converges? How can we be sure it has converged? If you don't run your algorithm for long enough, your answer will be very wrong.

Trace plots help. We also have convergence diagnostics.



You could throw away 6000 iterations as a **burn in** or **warm-up**.

Assessing convergence for scalars $\psi_{i,j}$ using $\hat{R}$

Run m chains for n iterations, $i = 1, \ldots, n$ and $j = 1 \ldots, m$.

$$\bar{\psi}_{\cdot\cdot} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \psi_{ij} \qquad \text{(overall average)}$$

$$\bar{\psi}_{\cdot j} = \frac{1}{n} \sum_{i=1}^{n} \psi_{ij} \qquad \text{(chain average)}$$

$$s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\psi_{ij} - \bar{\psi}_{\cdot j})^2 \qquad \text{(chain sd)}$$

$$W = \frac{1}{m} s_j^2 \qquad \text{(within-sequence variance)}$$

$$B = \frac{n}{m-1} \sum_{j=1}^{m} (\bar{\psi}_{\cdot j} - \bar{\psi}_{\cdot\cdot})^2$$

$$\widehat{\text{var}}^{+}(\psi \mid y) = \frac{n-1}{n} W + \frac{1}{n} B \qquad \hat{R} = \sqrt{\widehat{\text{var}}^{+}(\psi \mid y)/W}.$$

Split each chain into two, after discarding a burn-in/warm-up.

Section 2

Description of Algorithms

Metropolis-Hastings algorithm

At iteration $t-1$ you have θ^{t-1}. Propose $\theta^* \sim q(\theta \mid \theta^{t-1})$, and accept this draw with probability

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

If you accept, $\theta^t \overset{\text{set}}{=} \theta^*$. Otherwise, $\theta^t \overset{\text{set}}{=} \theta^{t-1}$.

Many algorithms are a special case of this one.

Metropolis-Hastings algorithm

Why it's widely-applicable:

$$
\begin{aligned}
a(\theta^{t-1}, \theta^*) &= \min\left\{1, \frac{p(\theta^* \mid y)q(\theta^{t-1} \mid \theta^*)}{p(\theta^{t-1} \mid y)q(\theta^* \mid \theta^{t-1})}\right\} \\
&= \min\left\{1, \frac{p(\theta^*, y)/p(y)q(\theta^{t-1} \mid \theta^*)}{p(\theta^{t-1}, y)/p(y)q(\theta^* \mid \theta^{t-1})}\right\} \\
&= \min\left\{1, \frac{p(y \mid \theta^*)p(\theta^*)q(\theta^{t-1} \mid \theta^*)}{p(y \mid \theta^{t-1})p(\theta^{t-1})q(\theta^* \mid \theta^{t-1})}\right\}.
\end{aligned}
$$

Don't need to know the normalizing constant/marginal likelihood/evidence!

Metropolis-Hastings algorithm

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

1. When is $a(\theta^{t-1}, \theta^*)$ nearly 1?
2. When is $a(\theta^{t-1}, \theta^*)$ nearly 0?
3. How should we pick $q(\theta^* \mid \theta^{t-1})$?
4. What is the ideal $q(\theta^* \mid \theta^{t-1})$?
5. What if $q(\theta^* \mid \theta^{t-1})$ is too peaked?
6. What if $q(\theta^* \mid \theta^{t-1})$ is too diffuse?

Metropolis-Hastings algorithm

Let's pick $q(\theta \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \sigma^2 \mathbf{I})$

`https://chi-feng.github.io/mcmc-demo/app.html#RandomWalkMH,banana`

Independent Metropolis Hastings

If $q(\theta \mid \theta^{t-1}) = q(\theta)$ (i.e. propose independently of past values), then you have the **Independent Metropolis Hastings** algorithm, which has acceptance probabilities

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

The proposals are iid, but the chain is Markovian!

Metropolis algorithm

If $q(\theta^* \mid \theta^{t-1}) = q(\theta^{t-1} \mid \theta^*)$ (i.e. q is **symmetric**), the acceptance probability

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

becomes

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

and they call this the **Metropolis** algorithm (drop the "Hastings").

The Gibbs Sampler

Say there are two parameters: $\theta = (\theta_1, \theta_2)$. The **Gibbs sampler** alternates between

1. $\theta_1^t \sim p(\theta_1 \mid \theta_2^{t-1}, y)$
2. $\theta_2^t \sim p(\theta_2 \mid \theta_1^t, y)$

If there are more parameters: $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$. The Gibbs sampler alternates between

1. $\theta_1^t \sim p(\theta_1 \mid \theta_{2:d}^{t-1}, y)$
2. $\theta_2^t \sim p(\theta_2 \mid \theta_1^t, \theta_{3:d}^{t-1}, y)$
3. $\vdots$
4. $\theta_d^t \sim p(\theta_d \mid \theta_{1:d-1}^t, y)$

This is only possible if you can sample from the **conditional posteriors** (i.e. need conditional conjugacy).

The Gibbs Sampler

Example on page 277:

1. $\theta_1 \mid \theta_2, y \sim \text{Normal}(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)$
2. $\theta_2 \mid \theta_1, y \sim \text{Normal}(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)$

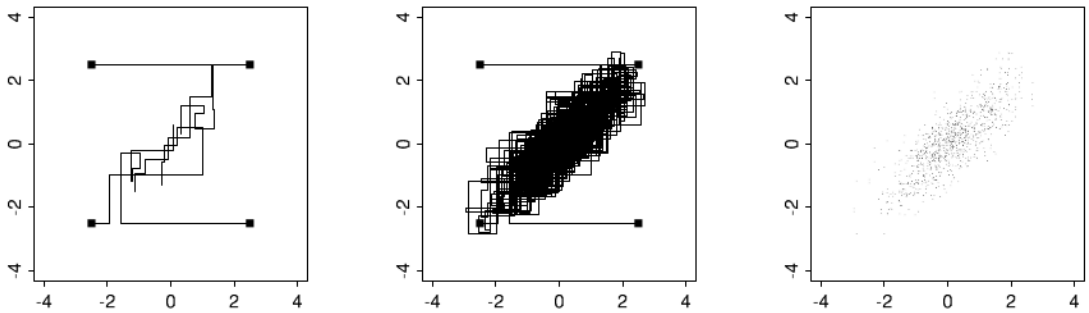


Figure 11.2 *Four independent sequences of the Gibbs sampler for a bivariate normal distribution with correlation $\rho = 0.8$, with overdispersed starting points indicated by solid squares. (a) First 10 iterations, showing the componentwise updating of the Gibbs iterations. (b) After 500 iterations, the sequences have reached approximate convergence. Figure (c) shows the points from the second halves of the sequences, representing a set of correlated draws from the target distribution.*

Section 3

A Closer Look

Metropolis-Hastings algorithm

Often $\theta^* \sim q(\theta \mid \theta^{t-1})$ is a density (continuous random variable), but the actual transition is not.

We call the probability of rejecting/staying put:

$$r(\theta^{t-1}) = 1 - \int a(\theta^{t-1}, \theta^*) q(\theta^* \mid \theta^{t-1}) \mathrm{d}\theta^*$$

Metropolis-Hastings algorithm

The transition dynamics for a MH chain is usually written down as a **kernel**

$$P(\theta^t \in A \mid \theta^{t-1}) = P(\theta^{t-1}, A)$$
$$= r(\theta^{t-1}) I(\theta^{t-1}, A) + \int_A a(\theta^{t-1}, \theta^*) q(\theta^* \mid \theta^{t-1}) \mathrm{d}\theta^*$$

1. it's more common for things to be written in a left-to-right format instead of a right-to-left
2. it's also more common for elements of the state space to be written with letters (e.g. x, y)

We switch to left-to-right, x and y notation, and then we switch back later.

Metropolis-Hastings algorithm

Call the target distribution π (i.e. the posterior), and call our Markov transition kernel $P(x, A)$.

π is the **stationary distribution** for P if

$$\int \pi(x)P(x, A)\mathrm{d}x = \pi(A).$$

Sometimes this is written as $\pi P = \pi$

Metropolis-Hastings algorithm

Call the target distribution π (i.e. the posterior), and call our Markov transition kernel $P(x, A)$.

The Markov chain is **reversible** if

$$\int_A \int_B \pi(x) P(x, dy) \mathrm{d}x = \int_B \int_A \pi(x) P(x, dy) \mathrm{d}x.$$

Being in B and then A has the same chances as being in A and then B.

This is the same thing as exchangeability! Think about when something like this wouldn't be true.

Metropolis-Hastings algorithm

Call the target distribution π (i.e. the posterior), and call our Markov transition kernel $P(x, A)$.

Reversibility implies stationarity. Just take $B = \mathsf{X}$ (the entire space):

$$\int_A \int_B \pi(x) P(x, dy) \mathrm{d}x = \int_B \int_A \pi(x) P(x, dy) \mathrm{d}x.$$

Why Everything "Works"

Virtually every MCMC we discuss has a "law of large numbers." This is because every Markov chain we talk about is **ergodic**, which means

$$\|P^n(x, A) - \pi(A)\|_{TV} \to 0 \qquad \text{as } n \to \infty.$$

This is convergence in total variation, which is stronger than convergence in distribution.

To calculate confidence intervals, you will need central limit theorems, which require stronger forms of ergodicity. These are more technical, and are outside the scope of this class.

Example: Hierarchical Normal Model

1. $y_{ij} \sim \text{Normal}(\theta_j, \sigma^2)$
2. $\theta_j \sim \text{Normal}(\mu, \tau^2)$
3. $p(\mu) \sim \text{Normal}(60, 100)$
4. $p(\tau^2) = \text{Inverse-Gamma}(.001, .001)$
5. $p(\sigma^2) = \text{Inverse-Gamma}(.001, .001)$

Example: Hierarchical Normal Model (MH)

We either need to worry about constraints, or we can sample on a transformed space.

Let's sample on the transformed space. Be careful of the Jacobians. If $p(\sigma^2) \propto (\sigma^2)^{-.0001-1} \exp\left[-.0001\sigma^{-2}\right]$, then

$$p(\log \sigma) \propto (\sigma^2)^{-.0001} \exp\left[-.0001\sigma^{-2}\right].$$

Also

$$p(\log \tau) \propto (\tau^2)^{-.0001} \exp\left[-.0001\tau^{-2}\right]$$

Nearly uniform, but they are proper, so they guarantee a proper posterior.

Example: Hierarchical Normal Model (MH)

See `hierarhical_normal_examples.R`

Example: Hierarchical Normal Model (Gibbs)

There are no tuning parameters required for Gibbs sampling, but you must derive the conditional posteriors.

1. $\theta_j \mid y, \mu, \tau^2, \sigma^2 \sim \text{Normal}\left(\frac{\bar{y}_{\cdot,j}(n_j/\sigma^2)+\mu(1/\tau^2)}{n_j/\sigma^2+1/\tau^2}, [n_j/\sigma^2 + 1/\tau^2]^{-1}\right)$
2. $\mu \mid \theta_{1:J}, \tau^2 \sim \text{Normal}\left(\frac{\bar{\theta}(J/\tau^2)+60*(1/100)}{J/\tau^2+1/100}, [J/\tau^2 + 1/100]^{-1}\right)$
3. $p(\tau^2 \mid \theta_{1:J}, \mu) = \text{Inverse-Gamma}(.0001 + J/2, .0001 + \frac{\sum_j(\theta_j-\mu)^2}{2})$
4. $p(\sigma^2 \mid y, \theta_{1:J}) = \text{Inverse-Gamma}(.0001 + N/2, .0001 + \sum_j \sum_i \frac{(y_{ij}-\theta_j)^2}{2})$

Example: Hierarchical Normal Model (MH)

See `hierarhical_normal_examples.R`