Chapter 3 MCMC Methods

Using this Chapter

This chapter provides an introduction to Markov Chain Monte Carlo (MCMC) methods and provides detailed discussion of the MCMC methods that have proved especially useful for marketing and micro-econometric problems. In addition, several key examples are developed which help set the stage for more complicated models covered in later chapters. These include the Gibbs samplers for binary probit, mixture of normals and hierarchical linear models. In addition, Metropolis methods are introduced and illustrated with the multinomial logit model. Those readers who desire an introduction to MCMC methods without much theoretical background should skip section 3.3 on Markov Chain theory and only skim the beginning of 3.9 on Metropolis methods (skip over the proof following the introduction of the "Continuous State Space Metropolis" in section 3.10).

Given a model (prior and likelihood) or set of models, the computational phase of Bayesian inference requires practical methods for summarizing/exploring the posterior distribution. In many cases, the posterior distribution is represented by an un-normalized density, $\pi^*(\theta)$, and the problem is to construct simulation-based estimates of various aspects of this distribution. For any problem outside the conjugate family, the posterior density will be of a form for which analytical results on marginals or moments will be unavailable. More importantly, problems in which the dimension of θ is much more than 100 can easily arise in marketing applications. For example, suppose we want to compute response to marketing instruments for each of 100 to 200 customers. We might formulate a group of 100 to 200 regression models each with 5 to 10 independent variables. If there are any linkages between these models through correlated unobservables[1], then we must explore a parameter space of dimension 500 to 2000. Problems of this dimension are considerably beyond the scope of importance-sampling methods. In order to tackle these problems, we will have to exploit

[1] The regression errors could be correlated (as in the SUR model of Zellner) or the prior on the regression coefficients could have correlations or dependencies across equations. The later is often formulated as a hierarchical model which is introduced below in section 3.7 and in chapter 5.

the structure of the model and introduce new methods for simulation from arbitrary distributions.

3.1 MCMC Methods

The idea behind MCMC methods is to formulate a Markov Chain on the parameter space. If care is taken to insure that this chain has $\pi(\ )$ as the equilibrium or "long-run" distribution of the chain, then the chain can be used to construct simulation-based estimates of the required integrals. Starting from some point in the parameter space, we simulate the chain forward. A sub-sequence of these draws can be used to construct simulation-based estimates of the posterior distribution of θ or any function of θ. For example, we can simply focus on one element of θ to simulate its marginal or we can average $h(\theta)$ over sequences of draws from the chain to estimate, $E_{\pi}\left[h(\theta)\right]$.

A Markov Chain specifies a method for generating a sequence of random variables $\left\{\theta_1,\theta_2,\ldots,\theta_{r,},\ldots\right\}$ starting from initial point θ_0 This sequence is created by specifying a way of transitioning or moving from θ_r to θ_{r+1}. Since we are dealing with random variables, this transition process is specified by choosing the conditional distribution, $\theta_{r+1}|\theta_r$, or $\theta_{r+1}|\theta_r \sim F(\theta_r)$. F denotes this conditional distribution which can be discrete, continuous or a mixture of discrete and continuous distributions. By iterating the conditional distribution forward we construct a joint distribution on the sequence. The fact that this conditional distribution only depends on the last θ (or that θ_r completely summarizes all information up to this point) is the Markov property which greatly simplifies simulation and analysis of the chain.

Clearly the Markov Chain was well-designed for simulation. To simulate from the MC,

```
START from $\theta_0$

DRAW        $\theta_1 \sim F(\theta_0)$

Replace $\theta_0$ with $\theta_1$ and REPEAT, a total of R times
```

This will create a set of realizations of the chain given the starting point. Under some conditions on the conditional distribution F, the distribution of $\theta_r | \theta_0$ will converge to a fixed and unique distribution as r goes to infinity. This distribution is called the stationary, invariant[2], or equilibrium distribution. If we can construct a Markov Chain with stationary distribution $\pi(\ )$ and the conditions for convergence are met, then we can use the MC method to construct a simulation method for estimating the posterior expectation of any function.[3]

$$E_\pi\left[h(\theta)\right] \doteq \tfrac{1}{R}\sum h(\theta_r) \tag{3.1.1}$$

Use of (3.1.1) is asymptotic in the sense that we are relying on the fact averages of the Markov Chain converge (again under some conditions) to the expectation under the stationary distribution.

$$\lim_{R\to\infty} \tfrac{1}{R}\sum h(\theta_r) = E_\pi\left[h(\theta)\right] \tag{3.1.2}$$

If a Markov Chain satisfies (3.1.2), it is called *ergodic.*

As a practical matter, we will be using large but finite R. Although the theory does not require it, most practitioners will discard some set of initial draws out of concern for the

[2] The meaning of the term "invariant" will be explained later in section 3.3.

effects of the "initial condition" or value of θ_0. This is sometimes called the "burn-in" period. The idea is that it will take a while for the chain to "equilibrate" or shrug off the effects of the initial condition. We then use the draws after the burn-in period to create simulation estimates. If we "burn-in" for B draws, then the MCMC estimate is given by

$$E_\pi\left[h(\theta)\right] \doteq \tfrac{1}{R-B}\sum_{r=B+1}^{R} h(\theta_r) \tag{3.1.3}$$

Of course, this is an example of a Monte Carlo integration estimate. The difference now is that the draw sequence is not an IID draw sequence. Since we are allowed to condition on the last value in generating the next value in a Markov Chain, the draws can be dependent. To some, this dependence is unfamiliar and troublesome. We are relying on something like the law of large numbers to insure the sample average converges to the "population" average. Most standard versions of the law of large numbers assume independence. As long as the dependence is not pathologically strong, then the intuition behind the law of large numbers still goes through.[4] We are still getting more information on the stationary distribution with every draw. The practical problem is that these simulation-based estimates may have large sampling errors.

We are relying on asymptotic theory to justify our use of simulation methods with large samples. That is, we are using the fact that long-run averages of draws from the Markov Chain will converge to the appropriate integral over the posterior distribution. More generally, the posterior distribution constructed from draws from the MC will closely approximate the true posterior distribution for large enough samples. This does not mean we are using sampling theory as the basis for our inferences regarding the data. The data is of fixed size, giving rise to a posterior for that sample. It is only when we approximate the

[3] Assuming, of course, that the posterior expectation of this function exists.

posterior that we appeal to asymptotics or long-run behavior. We should recognize that, up to computational limitations, the sample size used in any Monte Carlo method is under our control. In particular, we can increase the sample size as necessary and, for most problems, we can generate truly huge simulation samples at moderate computational cost. This means that the practical use of MCMC methods comes much closer to the long-run sampling experiments envisaged by the inventors of asymptotic theory than the usual application of these methods to small and fixed size samples.

The purpose of this introductory section was to give the reader an overview of the basic idea behind MCMC methods. We are a long way from practical application in the sense that we need to provide:

i. methods or algorithms for specifying chains with the right stationary distribution. This amounts to specifying the conditional distribution of $\theta_{r+1}|\theta_r$ using information about $\pi(\ )$.

ii. Theoretical assurance that the methods in i) will produce ergodic chains.

iii. Practical guidance on convergence including some notion of how long we should run the chain and how long the "burn-in" period should be.

There is good news and bad news regarding the answers to these questions. The good news is that there are families of algorithms which can deliver chains for arbitrary posterior distributions. Further, these algorithms enjoy very strong theoretical convergence properties under relatively mild and verifiable conditions. The "bad" news is that even a theoretically convergent chain may be very slow to converge and/or highly dependent. This means that some care and experimentation must be used in the selection and use of the MCMC

[4] Those trained in time series analysis will recall that there are versions of the Law of Large Numbers for dependent but stationary sequences.

algorithms. However, it is fair to say that MCMC methods have been applied to problems of dimension (exceeding 1000) and complexity (problems for which the likelihood is intractable) well beyond the original developers' wildest dreams.

3.2 A Simple Example: Bivariate Normal Gibbs Sampler

The ideas introduced in 3.1 can be better understood by considering one of the most useful and well-used MCMC methods, the Gibbs Sampler. We will treat the general case of the Gibbs sampler in 3.4 and we will see many non-trivial examples. However, it is best to start with a very simple problem.

Consider the problem of simulating from the bivariate normal distribution.

$$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} \sim N\left(\begin{pmatrix}0\\ 0\end{pmatrix}, \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right) \tag{3.2.1}$$

In section 2.10.4, the standard method for drawing from the multivariate normal distribution was presented. Recall that we only have to compute the Cholesky root of the covariance matrix and use this to induce the proper level of correlation. Let Z be an R x 2 matrix of iid N(0,1) draws, then we simply compute

$$\Theta = ZL' \quad \text{where } L = \begin{bmatrix}1 & 0\\ \rho & \sqrt{1-\rho^2}\end{bmatrix} \tag{3.2.2}$$

The R x 2 matrix, Θ, is a matrix whose rows are iid draws from (3.2.1).

The Gibbs sampler specifies a Markov Chain whose stationary distribution is the bivariate normal. The transition mechanism in the Gibbs sampler is specified through iterative sampling from conditional distributions. At point $\theta_r = \begin{pmatrix}\theta_{r,1}\\ \theta_{r,2}\end{pmatrix}$ (here $\theta_{r,i}$ is the rth draw of component i), the next random variables $\begin{pmatrix} \theta_{r+1,1} \\ \theta_{r+1,2} \end{pmatrix}$ are constructed by drawing from the two conditional distributions associated with the bivariate normal.

$$\theta_2 \Big| \theta_1 \sim N\left(\mu_2 + \rho \frac{\sigma_2}{\sigma_1}(\theta_1 - \mu_1), \sigma_2^2\left(1-\rho^2\right) \right)$$

and

$$\theta_1 \Big| \theta_2 \sim N\left(\mu_1 + \rho \frac{\sigma_1}{\sigma_2}(\theta_2 - \mu_2), \sigma_1^2\left(1-\rho^2\right) \right)$$

Example of BIVARIATE NORMAL GIBBS SAMPLER:

Start at point θ_0

Draw θ_1 in two steps

$$\begin{aligned} \theta_{1,2} &\sim N\left(\rho\theta_{0,1}, 1-\rho^2\right) \\ \theta_{1,1} &\sim N\left(\rho\theta_{1,2}, 1-\rho^2\right) \end{aligned} \tag{3.2.3}$$

Repeat as long as desired to draw $\theta_2 \big| \theta_1, \ldots, \theta_{r+1} \big| \theta_r, \ldots$

We draw first from the conditional distribution of the second component given the previous draw of the first component. We then draw the first component given the most recent drawn value of the second component. This means that we move from one point to another in the two-dimensional parameter space in a sequence of two moves along the coordinate axes. In this particular sampler, we draw $\theta_{r+1,2} \big| \theta_{r,1}$ and then $\theta_{r+1,1} \big| \theta_{r+1,2}$. This means that we only "use" the initial value $\theta_{0,1}$ and do not use $\theta_{0,2}$ (check the code!).

Of course, we could have swapped the order of these "intermediate" steps and still have a valid (but different) sampler.

This simple example shows that one of the problems with MCMC algorithms is that they don't lend themselves to vectorization that is required for the most efficient use of interpreted languages such as R. There is no avoiding the "loop" here since the arguments in the loop are recursively dependent and cannot be computed prior to vector operations. This means that there is at least one "loop" even in the most efficient R code to implement MCMC methods. Our experience, however, has shown that this is not a serious computational constraint for many problems of interest.

Figure 3.1 illustrates this algorithm for $\rho = .9$ and a starting value $\begin{pmatrix} \theta_{0,1} \\ \theta_{0,2} \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$. To move away from the initial condition, we first draw the second (vertical) component value. Given that we start from the first component value of 2, we draw (according to (3.2.3)) from a $\mathrm{N}(.9 \times 2, .19)$ distribution, illustrated by the vertically oriented density curve in Figure 3.1. This distribution is centered at 1.8 but with a fairly large standard deviation of .44. A realized value is shown on the figure by the "•" symbol. This realized value is about 1.7, near the mean of the conditional distribution. We now draw the first component given the second value of 1.7 by drawing from a $\mathrm{N}(.9 \times 1.7, .19)$ distribution. These two draws combine to move the chain from the initial value of (2,-2) to a new value of approximately (1, 1.7). This move is given by the dotted arrow on the figure.

Figure 3.2 shows a realization of the first 20 draws from the same initial condition as in figure 3.1. The figure shows both the "intermediate" moves internal to the sampler (that is, the component by component updatings) as well as the final result. Each of these points are connected by lines to trace out the movements. The fact the intermediate moves are included is the reason why the trace consists of only vertical and horizontal line segments of random length. The chain moves quickly away from the initial condition to the region where the bivariate normal has substantial mass. The reason for this is that the initial value of (2, -2) is highly unusual for this bivariate normal. The conditional distributions capture this immediately by moving the second component to a value more in line with the strong positive correlation.

The average size of the move along any given coordinate axis is constant since the standard deviation of normal conditional distributions, $\sqrt{1-\rho^2}$, does not depend on the conditioning argument. Clearly as ρ gets even closer to one in magnitude, the size of the moves of the sampler will be very limited. However, for all but extremely large values of ρ , the chain will dissipate the effects of the initial conditions rapidly although it may move slowly once it gets to the areas of high mass. After only 40 draws, the chain has started to navigate the regions where the bivariate normal distribution puts mass, but nowhere near adequately for the purpose of estimating probabilities of sets or moments. For example, we certainly would be foolish to estimate the probability of a set by simply computing the proportion of draws falling in this set. The theory of MCMC would tell us that as the number of draws tends to infinity this answer can be computed to any desired degree of accuracy. Figure 3.3 shows 1000 draws (without the intermediate moves on the lattice) of the sampler. Clearly, the chain seems to navigate freely and the regions of high mass are dark with ink.

Figure 3.2 also shows that there is substantial dependence for replicates close together in the sequence. If the chain is in the positive orthant along the “ridge” created by the positive correlation, then it may take several (or more) moves to navigate into the negative orthant. One way of measuring this serial or "time" dependence is to compute the sample autocorrelation function.

$$(3.2.4) \qquad s_{\theta_1}(k) = \frac{\sum_{r=k}^{R}\left(\theta_{r,1} - \bar{\theta}_1\right)\left(\theta_{r-k,1} - \bar{\theta}_1\right)}{\sum_{r=1}^{R}\left(\theta_{r,1} - \bar{\theta}_1\right)^2}$$

The top panel of figure 3.4 shows the autocorrelation function for the first component of θ. The lag or "order" of the autocorrelation is the number of "periods" or draws separating random variables in the chain (k in (3.2.4)). As we might expect, there is a fair amount of autocorrelation in the sequence of draws. However, by lag 10 or 12, there is no appreciable dependence.

If this sampler is ergodic, we can use sample averages of functions evaluated on the chain draws to estimate the expectations of those functions with respect to the stationary distribution of the chain. Figure 3.3 suggests that the stationary distribution for this chain is actually the bivariate normal distribution. We can use the sample correlation of the draws of θ_1 and the draws of θ_2 to estimate the "population" correlation or the expectation of the function $h(\theta) = \theta_1\theta_2$ with respect to the bivariate normal. In this example, we know that $E\left[h(\theta)\right] = E\left[\theta_1\theta_2\right] = \rho$. The bottom panel shows the sample correlation based on the Gibbs sampler draw sequence (dotted line) for samples of successively larger size starting from the beginning of the chain. For example, the value of the line corresponding to the horizontal axis value of 100 is the sample correlation based on replicates 0,1, …, 100. The figure illustrates the ergodicity of this chain. These sample averages rapidly converge to the true value of .9. The figure also illustrates the importance of a "burn-in" period. It takes a good 100 or so iterations to "work off" the effects of our extreme initial condition. Of course, with modern algorithms for normal random number generators, we can generate in excess of 50 million univariate normals per second on garden variety PCs. This means we can run out this sampler to a million or more draws in less time than it takes us to graph the results. However, the problems of dissipation of initial conditions and serial dependence demonstrated for this example can be found in higher dimensional situations even with tens or even hundreds of thousands of draws.

The serial dependence in the Gibbs sampler draws is the price of this method. The estimates of integrals using these dependent draws can be substantially less efficient (in the sense of sampling error) than estimates based on iid draw sequences. For the bivariate normal, we have an iid sampler and the solid line in the bottom panel of 3.4 shows the convergence of estimates of ρ based on a sequence of iid draws. These estimates converge much more rapidly than thosed based on the Gibbs sampler draws. This, of course, is not really a fair comparison since the Gibbs Sampler is used on problems for which iid draw algorithms are not available or are computationally infeasible.

The Gibbs Sampler for the bivariate normal problem is a nice illustration of the general idea of MCMC as well as an introduction to one important method. However, this method exploits the very special structure of the normal distribution and the ease with which the conditional distributions can be drawn from. It remains to be seen how generalizable this approach is and what alternatives are available if a strict Gibbs Sampler cannot easily be constructed.

3.3 Some Markov Chain Theory

Before outlining some of the more useful algorithms for constructing Markov chains, we will discuss some of the basic theory. While Markov chains are easy to invent, analysis of the convergence and distributional properties of these chains can be involved. This literature has a complex notation and requires at least some familiarity with measure-theoretic probability for full access. Tierney (1994) and Robert and Casella (2004, Chapters 7 and 10) have distilled much of the relevant theory from this literature. However, both Tierney and Robert and Casella still require measure theory and a considerable time investment to digest. One view is that theory is largely irrelevant for the practitioner and, therefore, one should only provide a menu of algorithms along with assurances that they will work. Our view is that a practitioner should understand the basic intuition as to why his methods work. This basic intuition can help diagnose algorithmic and programming errors. In addition, some notation and vocabulary may make it easier to follow the practical implications of the burgeoning MCMC literature, some of which is highly technical. For this reason, we provide a minimal set of concepts and illustrate these with discrete and continuous state space chains.

In most instances, the parameters in models (remember this includes all unobservables) will be continuous random variables and $\pi(\ )$ will be a standard density. Thus, most of the Markov chains we will consider generate random variables with a continuous component in their transition distribution and the chain navigates in some subset of $\Re^k$. This sort of Markov chain is called a continuous state space Markov chain. We will start, however, by considering discrete state space[5] chains. Much of the intuition we develop for discrete state space chains carries over to the continuous case with some technical difficulties which we will note.

Interest in discrete state space chains can be motivated by considering a discrete approximation to the posterior distribution. We could lay a grid down along each of the coordinate axes and, therefore, construct a discrete approximation to the posterior distribution using the heights of the posterior density on this grid of points. Let g_i be a grid of values $\left(g_{i,1},\ldots g_{i,m}\right)$ for the ith component of the parameter vector, θ, where m is the number of grid points. If we lay a grid of points on each of the k coordinate axes in the parameter space, Θ, then we have constructed the product set, $G = g_1 \times g_2 \times \ldots g_k$. G has m^k elements in it. An element $\theta \in G$ takes on one of the values of the grid for each of the k axes, $\theta_{i_1,i_2,\ldots,i_k}{}' = \left(g_1\left(i_1\right), g_2\left(i_2\right),\ldots,g_k\left(i_k\right)\right)$. i_j is the index of the grid for θ component j and $g_d\left(i_d\right)$ is the i_d value of the g_d grid at this index. For example, consider the two dimensional case, $\theta_{2,3}$ is value corresponding to the 2nd element of the grid on the first coordinate and the 3rd element of the grid on the second component. $\theta_{2,3}{}' = \left(g_1(2), g_2(3)\right)$; as both indices range over the m possible values, all possible discrete values for θ are enumerated.

Figure 3.5 illustrates discretization for the two dimension case with a continuous bivarate $\pi(\ )$. In the left panel, the bivariate double exponential density is plotted, $\pi(\theta) \propto \exp\left(-|\theta_1| - |\theta_2|\right)$. This density has a mode at (0, 0) and a scale of 1. If we lay down an equal spaced grid of 10 values between -2 and 2 on each coordinate axis, we have the basis for a 10 x 10 or 100 point discretization of $\pi(\ )$. The right panel of figure 3.5 shows the discrete approximation to the density based on this grid. A simple discrete approximation would be to normalize the 100 values of π on the grid.

We have seen that we can motivate an interest in discrete state space Markov chains by discretizing the parameter space. If grids of m points are used for each component of θ,

[5] The term state space itself conjures up a discrete world in which the chain can only take on a finite or, at least countable, number of values or "states." The current "state" of the chain is nothing more than the current

then any Markov Chain would be defined on a state space with m^k elements. This grid is the basis for a discrete approximation to π and an IID sampler from this discrete approximation.

Discrete Approximation Algorithm

Lay down grids on each axis, $(g_1, g_2, \ldots, g_k)$

Develop a mapping[6] from the integers $j = 1, 2, \ldots, m^k$ to each of the m^k points in $G = g_1 \times g_2 \times \ldots g_k$, $i_1(j), \ldots, i_k(j)$

Evaluate the un-normalized posterior[7], π^*, at each of the m^k grid points and normalize this vector

$$p_j = \left(\pi^*\left(\theta_{i_1(j),\ldots,i_k(j)}\right)\right) / \Sigma_{l=1}^{m^k} \pi^*\left(\theta_{i_1(l),\ldots,i_k(l)}\right)$$

θ ~ Multinomial with probabilities given abov.e

This seems like a very appealing general idea to construct IID samplers. In particular, it would work extremely well for the two dimensional example in figure 3.5 and a grid of 100 points on each axis. With k=10 and a grid of 100 points, we would have to make 10 billion evaluations of the posterior density. For simple densities and small datasets, this is very feasible on modern 1 G FLOP computers. However, for complicated densities or larger dimension this quickly becomes a computational nightmare since the computations required rise to the power of k. Moreover, there is an assumption that we would know where

realized value which corresponds to one of the finite number of possibilities that define the state space.

[6] Usually this is done by establishing an order in which the subscripts denoting the components of θ are allowed to vary and then considering a base m representation of the integer j which gives the grid elements for each of the k components as digits.

[7] Not only is it unnecessary to evaluate the normalized posterior but to the extent that any normalizing constants are expensive to compute, this could be very inefficient. Recall that the normalizing constants can be functions of the data, making this a real possibility.

$\pi(\ )$ concentrates its mass. Since the point of doing MCMC in the first place is to explore the posterior and see where its mass is, it may be difficult to lay down grids so that there is enough detail where the posterior is concentrated. Even if the posterior is uni-modal, we don't know much about its spread and shape except by resort to asymptotic approximations.

Even on discrete parameter spaces, we need methods of exploring that are capable of handling high dimensional problems. Markov Chains are one class of these methods. We now introduce general notation for a discrete space Markov Chain. Let $S=\left\{\theta^1,\theta^2,\ldots,\theta^d\right\}$ be the state space, we define a Markov Chain as the sequence of random variables, $\left\{\theta_{1,}\theta_2,\ldots,\theta_r,\ldots\right\}$ given θ_0 generated by the following transition.

(3.3.1) $$\Pr\left[\theta_{r+1}=\theta^j \middle| \theta_r=\theta^i\right]=p_{i,j}$$

Given the current realization of the chain, the rows of the matrix, P, formed from p_{ij} specify the conditional distribution of the chain at the next iteration. The Markov property states that conditional distribution of θ_{r+1} depends only on θ_r and nothing from the "earlier" history of the chain Up to this point, we have conditioned on a specific initial value θ_0. This is certainly how we simulate chains. One way of thinking about the chain is how this initial value is transformed. However, in analyzing the behavior of Markov chains, it is more useful to consider the more general case in which we specify a distribution for the initial value, $\theta_0\sim\pi_0$. By convention in the Markov Chain literature, distributions over states are denoted by the *row* vector of probabilities. The chain transforms this distribution into a new distribution in each iteration of the chain. Consider the distribution of θ_1 given that $\theta_0 \sim \pi_0$.

$$\Pr\left[\theta_1 = \theta^j\right] = \sum_{i=1}^{d} \Pr\left[\theta_0 = \theta^i\right] p_{ij} = \sum_{i=1}^{d} \pi_{0,i} p_{ij}$$

In matrix form, the above equation states that $\pi_1 = \pi_0 P$. After r iterations, we have $\theta_r \sim \pi_0 P^r$.

As the number of iterations increases, we might expect that the effects of the initial distribution π_0 will "wear off." In addition, we might expect some chains to "settle down" to some sort of equilibrium distribution. Here we are ruling out chains that have "absorbing" states or sets of states that they get trapped in and never get out of or, the converse, that they never visit some states (these are called "reducible" chains). If $p_{ij} > 0$ for all i and j, then all states will communicate with one and other and there can be no subset of states to get trapped in. If you get into state i, there is some positive probability that you will get out of it. However, we can immediately see the distinction between the theory and practice. If p_{ij} is small for all i and a specific j, then the chain might only visit state j very infrequently in a finite sequence of draws[8] even though, in theory, this state will be visited infinitely often!

If $p_{ij} > 0$ for all i,j, then the chain is called *irreducible* and there exists a stationary distribution, π, such that

(3.3.2) $$\lim_{r \to \infty} \pi_0 P^r = \pi$$

[8] Resulting in a very poor estimate of the marginal probability of state j.

(3.3.2) states that, if we start from any distribution, we will get to π eventually. If we start in π, then we must stay in π otherwise π would not be the stationary distribution.

(3.3.3)
$$\pi P = \pi$$

(3.3.3) is the reason that the stationary distribution is also called the *invariant* distribution. For discrete state space chains, irreducibility also implies ergodicity (chain averages of functions converge to their expectation under π).

If presented with a discrete Markov chain that claims to have π as the stationary distribution, it is straightforward to check that (3.3.3) holds. However, it will turn out that for more general state space chains, it will be useful to have an equivalent property called *time reversibility*. Time reversibility states that, if we reverse the sequence order of a Markov chain, the resulting chain will have the same transition behavior. First, we will reverse the order of the chain and check to see that it is still Markov. Then we will compute the transition probabilities for the reversed chain in terms of the standard forward chain. We want to compute the probability of being in state j at "time" r given the future history.

Using the standard definition of conditional probability, we can write the "backward" transition probability as follows:

$$\Pr\left[\theta_r = \theta^j \middle| \theta_{r+1} = \theta^{i_1}, \theta_{r+2} = \theta^{i_2}, \ldots, \theta_{r+s} = \theta^{i_s}\right]$$
$$= \frac{\Pr\left[\theta_r = \theta^j, \theta_{r+1} = \theta^{i_1}, \theta_{r+2} = \theta^{i_2}, \ldots, \theta_{r+s} = \theta^{i_s}\right]}{\Pr\left[\theta_{r+1} = \theta^{i_1}, \theta_{r+2} = \theta^{i_2}, \ldots, \theta_{r+s} = \theta^{i_s}\right]}$$

To see that the "reversed" or "backwards" chain is Markov, we can write this ratio using terms which involve the future r+2 periods and beyond and other terms which involve only the rth and (r+1)st period.

$$=\frac{\Pr\left[\theta_r=\theta^j\right]\Pr\left[\theta_{r+1}=\theta^{i_1}\middle|\theta_r=\theta^j\right]\Pr\left[\theta_{r+2}=\theta^{i_2},\ldots,\theta_{r+s}=\theta^{i_s}\middle|\theta_r=\theta^j,\theta_{r+1}=\theta^{i_1}\right]}{\Pr\left[\theta_{r+1}=\theta^{i_1}\right]\Pr\left[\theta_{r+2}=\theta^{i_2},\ldots,\theta_{r+s}=\theta^{i_s}\middle|\theta_{r+1}=\theta^{i_1}\right]}$$

The Markov property of the forward chain implies that

$$\Pr\left[\theta_{r+2}=\theta^{i_2},\ldots,\theta_{r+s}=\theta^{i_s}\middle|\theta_r=\theta^j,\theta_{r+1}=\theta^{i_1}\right]=\Pr\left[\theta_{r+2}=\theta^{i_2},\ldots,\theta_{r+s}=\theta^{i_s}\middle|\theta_{r+1}=\theta^{i_1}\right].$$

The conditional probabilities for periods r+2, …, r+s, cancel from the numerator and denominator. This means that the reversed chain is also Markov and, further, that

$$\Pr\left[\theta_r=\theta^j\middle|\theta_{r+1}=\theta^{i_1},\theta_{r+2}=\theta^{i_2},\ldots,\theta_{r+s}=\theta^{i_s}\right]=\Pr\left[\theta_r=\theta^j\middle|\theta_{r+1}=\theta^{i_1}\right]$$
$$=\frac{\Pr\left[\theta_r=\theta^j\right]\Pr\left[\theta_{r+1}=\theta^{i_1}\middle|\theta_r=\theta^j\right]}{\Pr\left[\theta_{r+1}=\theta^{i_1}\right]}$$

If P^* represents the transition matrix of the reverse chain, then the above equation is the relationship

(3.3.4)
$$p_{ij}^*=\frac{\pi_j p_{ji}}{\pi_i}$$

Time reversibility requires that $p_{ij}^*=p_{ij}$. This means that time reversibility is equivalent to

(3.3.5)
$$p_{ij}=\frac{\pi_j p_{ji}}{\pi_i}\ \text{ or }\ \pi_i p_{ij}=\pi_j p_{ji}$$

Roughly speaking, the property of time reversibility implies that chance of seeing a transition from state i to state j is the same as the chance of seeing a transition from state j to state i. Some say that (3.3.5) means that the chain described by P is "reversible with respect to π."

There is a complete equivalence between reversibility and the stationarity of π in the sense that if a chain is reversible with respect to some distribution ω then ω is also the stationary distribution of the chain. Reversibility wrt to ω means $\omega_i p_{ij}=\omega_j p_{ji}$. Summing both sides over i, we obtain

$$\sum_i \omega_i p_{ij} = \sum_i \omega_j p_{ji} = \omega_j \sum_i p_{ji} = \omega_j \times 1$$

or

$$\omega P = \omega$$

and ω is the stationary distribution of the chain.

The bivariate normal Gibbs sampler discussed in section 3.2 is an example of a continuous state space chain. It will, therefore, be important to extend the ideas we have developed for discrete state space chains to the continuous case. Fortunately, the basic ideas of reversibility and invariant distributions extend without much difficulty. There are some technical difficulties in establishing convergence and ergodic results but central intuition that the chain must freely navigate is at the core of these results just as in the discrete case.

In the continuous state space case, the transitional conditional distribution of $\theta_{r+1} | \theta_r$ must have a continuous component. Rather than specifying this distribution via the probabilities of each of the singletons $\left\{\theta^i\right\}$ that comprise the state space, we must specify the conditional distribution by associating probabilities with sets in the state space. For example, consider a set $A \in \Theta$,[9] then the chain is specified by the probabilities of the set A given the value of the chain on the previous iteration. This is sometimes called the Kernel of the chain. $K(\theta, A)$ is the probability of set A given the chain is at value θ. Some Kernels (such as the one corresponding to the Gibbs Sampler, but not all Kernels) can be represented using a standard density.

$$K(\theta, A) = \int_A p(\theta, \vartheta) d\vartheta$$

[9] Technically, we can only assign probabilities to certain subsets, but we will gloss this over.

$p(\theta,\vartheta)$ is a density for fixed θ. To distinguish p from K, we will call p( ) the transition function of the Kernel.

Analogous to the discrete case (see (3.3.3)), we can defin the concept of an invariant distribution. A distribution with density $\pi(\ )$ is an invariant distribution if the probability of $A \in \Theta$ computed under $\pi(\ )$ is the same as one-step ahead probability of A given that $\theta \sim \pi$.

$$\int_A \pi(\theta)d\theta = \int_\Theta K(\theta,A)\pi(\theta)d\theta = \int_\Theta \left[\int_A p(\theta,\vartheta)d\vartheta\right]\pi(\theta)d\theta$$

The principle of time reversibility can also be defined for the continuous state space chain. The chain is said to be reversible with respect to a distribution with density $\omega(\theta)$ if the transition function satisfies

(3.3.6)
$$\omega(\theta)p(\theta,\vartheta) = \omega(\vartheta)p(\vartheta,\theta)$$

If (3.3.6) is satisfied, then the stationary distribution of the chain with transition function $p(\bullet,\bullet)$ has density $\omega(\theta)$.

$$\begin{aligned}\int_\Theta \omega(\theta)K(\theta,A)d\theta &= \int_\Theta \omega(\theta)\int_A p(\theta,\vartheta)d\vartheta d\theta \\ &= \int_\Theta\int_A \omega(\theta)p(\theta,\vartheta)d\vartheta d\theta\end{aligned}$$

Reversing the order of integration and using the reversibility condition,

$$\begin{aligned}&= \int_A\int_\Theta \omega(\vartheta)p(\vartheta,\theta)d\theta d\vartheta = \int_A \omega(\vartheta)\left[\int_\Theta p(\vartheta,\theta)d\theta\right]d\vartheta \\ &= \int_A \left(\omega(\vartheta)\times 1\right)d\vartheta\end{aligned}.$$

ω, therefore, is the invariant distribution of the chain. Thus, reversibility and invariance are equivalent.

Finally, the concept of *irreducibility* can be extended to the continuous state space setting. Irreducibility requires that the chain navigate the state space freely so that it cannot get trapped in a subset of the entire state space. In the discrete case, all that is required is strict positivity of the transition probabilities, P > 0. In the continuous case, the definition of irreducibility is straightforward but verification that a given Kernel produces an irreducible chain is not always a simple matter. A chain with kernel, K, is irreducible with respect to $\pi(\;)$ if every set A with positive π probability can be reached with positive probability after a finite number of steps. $\int_A \pi(\theta)d\theta > 0 \Rightarrow$ there exists $n \geq 1$ such that $K^n(\theta, A) > 0$.

3.4 Gibbs sampler

The Gibbs sampler is a Markov chain obtained by cycling through a set of conditional distributions of π. If we break θ into p separate "groups" or "blocks" of parameters, then the Gibbs sampler is defined by iterative sampling from each of these p conditional distributions.

GIBBS SAMPLER

(3.4.1) $\theta' = (\theta_1, \theta_2, \ldots, \theta_p)$ (p groups or blocks[10])

Set θ_0

Sample from

[10] We will see examples where each block is only one dimensional and others in which each block corresponds to subsets of the complete set of unobservables.

$$
(3.4.2) \qquad \begin{aligned}
&\theta_{1,1} \sim f_1\left(\theta_1 \middle| \theta_{0,2},\ldots,\theta_{0,p}\right) \\
&\theta_{1,2} \sim f_2\left(\theta_2 \middle| \theta_{1,1},\theta_{0,3},\ldots,\theta_{0,p}\right) \\
&\quad\vdots \\
&\theta_{1,p} \sim f_p\left(\theta_p \middle| \theta_{1,1},\ldots,\theta_{1,p-1}\right)
\end{aligned}
$$

to obtain the 1st iterate

REPEAT as necessary

$f_1,\ldots,f_p$ are the appropriate conditional densities derived from π.

$$f_i = \pi(\theta) \big/ \int \pi(\theta)\, d\theta_{-i}\ , \quad \theta_{-i}^{'} = \left(\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_p\right)$$

Implementation of the Gibbs sampler requires the ability to sample from the set of conditional posterior distributions. In many situations, it is possible to define the groups or blocks of parameters so that the conditional distributions are of known form and can be sampled from very efficiently using the algorithms defined in chapter 2. As a "fall-back" or default alternative, one could always define a Gibbs sampler based on the k univariate conditionals implied by π. This would require a generic method for making draws from univariate distributions whose un-normalized density can be evaluated.

The Gibbs sampler defined by (3.4.2) is clearly a Markov Chain. It is also easy to verify that the invariant distribution of this chain is π. To see this, consider the bivariate case. $\theta^{'} = \left(\theta_1,\theta_2\right)$. The rth iteration of the bivaraite Gibbs Sampler draws successively from two conditional distributions.

$$
(3.4.3) \qquad \begin{aligned}
&\theta_{r+1,2} \sim \pi_{2|1}\left(\theta_{r,1}\right) \\
&\theta_{r+1,1} \sim \pi_{2|1}\left(\theta_{r+1,2}\right)
\end{aligned}
$$

To check that π is the invariant distribution, we must verify that if $\theta_r \sim \pi(\;)$, then $\theta_{r+1} \sim \pi(\;)$. The notation $\pi_{i|j}(\theta)$ means the distribution of component i given that component j takes on the value θ.

Suppose $\theta_r \sim \pi(\;)$. This means that $\theta_{r,1} \sim \pi_1(\theta_1) = \int \pi(\theta_1, \theta_2) d\theta_2$. $\theta_{r+1,2}$ is a draw from the conditional distribution $\pi_{2|1}$. Therefore, the distribution of $\theta_{r+1,2}$ from one iteration of the sampler is a draw from the marginal distribution of the second component, $\theta_{r+1,2} \sim \pi_2 = \int \pi_{2|1}(\theta_2 | \theta_1) \pi_1(\theta_1) d\theta_1$. The same argument can be used to show that $\theta_{r+1,1} \sim \pi_1$ using the fact that $\theta_{r+1,2} \sim \pi_2$. Thus, the (r+1)st iteration of the chain reproduces a draw from the invariant distribution.

Convergence of the Gibbs sampler is assured under very mild conditions.[11] If the Gibbs sampler is irreducible,[12] then Theorem 1 of Tierney (1994, p. 1712) assures convergence of the n step ahead distribution to the invariant distribution for almost all starting points. Most examples of reducible Gibbs Samplers involve sort of constraint on the state space, Θ.[13] If the state space is the Cartesian product of intervals on each coordinate axis, then a Gibbs sampler whose conditional densities are strictly positive everywhere and whose marginal densities exist will be irreducible.[14] Furthermore, by Tierney Theorem 2, Corollary 1, this sampler will converge to the stationary distribution

[11] These conditions are so mild that some, c.f. Liu (2001), assume convergence and focus attention on the rate of convergence, extent of autocorrelation and various ideas for improving performance.

[12] Also, it is required that the sampler be aperiodic. Periodic Chains require deterministic constraints on movement of the chain, something not found in MCMC algorithms.

[13] The classic example involves two disjoint disks located on the 45 degree line. The sampler would get "trapped" in one disk depending on the initial condition. A closely related is the example in Hobert et al (1997) in which the state space consists of two boxes that are oriented along the 45 degree line and are tangent at one vertex. In this example, the state space is a subset of a product set which defines a larger box enclosing the two smaller boxes. Finally, an example in Geweke(2003) has a state space consisting of a solid polygon with an acute angle for one vertex and again, oriented on the 45 degree line. In this example, this vertex is an

from all initial points and will be ergodic. This justifies the practical use of the Gibbs sampler – to start from an arbitrary initial condition and use sample averages to approximate integrals of the posterior. Marginal densities will not exist for improper posteriors. If proper priors are used and if the likelihood is bounded, we can avoid the problem of improper posteriors and Gibbs Samplers which attempt to approximate quantities that don't exist! We regard this as yet another (and not even the most important) reason to avoid improper, "reference" or "diffuse" priors.

An appreciation for the power of the Gibbs sampler as well as better feel for implementation can be obtained by considering some important and nontrivial examples. We will consider first the generalization of the Multivariate Regression Model introduced in chapter 2.

3.5 Gibbs Sampler for the SUR Regression Model

In the SUR model, a system of m regression equations are related through correlated error terms.

$$\begin{gathered} y_i = X_i\beta_i + \varepsilon_i \\ \left(\varepsilon_{k,1}, \varepsilon_{k,2}, \ldots, \varepsilon_{k,m}\right) \sim N\left(0, \Sigma\right) \\ i = 1, \ldots, m \quad k = 1, \ldots, n \end{gathered} \tag{3.5.1}$$

It will be convenient to stack up the m regressions in (3.5.1) into one large regression.

$$y = X\beta + \varepsilon \quad \varepsilon \sim N\left(0, \Sigma \otimes I_n\right) \tag{3.5.2}$$

with

absorbing state. Again, the product set rules this example out while keeping what is needed for virtually all applications.

[14] This is equivalent to the positivity condition (c.f. Robert and Casella (2004), p. 345).

$$y' = (y_1, \ldots, y_m) \quad X = \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_m \end{bmatrix} \quad \beta' = (\beta_1, \ldots, \beta_m) \quad \varepsilon' = (\varepsilon_1, \ldots, \varepsilon_m)$$

As discussed in 2.8, there is no convenient natural conjugate joint prior on $\{\beta_i\}, \Sigma$. Recall that the natural conjugate prior for the Multivariate Regression Model (MRM) has the prior on β depending on Σ. This prior embodies the notion that information on β can never be *scale-independent*. In all three situations (SUR, MRM and univariate regression), we may have prior information which is non data-based and, hence, would not always be scale dependent. A simple prior specification would be to make β and Σ a priori independent.

$$\begin{aligned} &p(\beta, \Sigma) = p(\beta)p(\Sigma) \\ &\beta \sim N\left(\bar{\beta}, A^{-1}\right) \\ &\Sigma \sim IW(\nu_0, V_0) \end{aligned} \tag{3.5.3}$$

These priors are not conjugate but they are *conditionally* conjugate. Given Σ, the SUR likelihood in (3.5.2) can be written in the standard normal regression form and is conjugate with a normal prior on the stacked vector of regression coefficients. Given β, the SUR likelihood is in a form that has an IW conjugate prior. The Gibbs sampler simply alternates between draws from these two sets of conjugate distributions.

Given Σ, we can transform (3.5.2) into a system with uncorrelated errors using root of the cross-equation covariance matrix. $\Sigma = LL'$ and $L^{-1}\Sigma\left(L^{-1}\right)' = I_m$. This means that, if we premultiply both sides of (3.5.2) by $L^{-1} \otimes I_n$, the transformed system has uncorrelated errors

$$\tilde{y} = \tilde{X}\beta + \tilde{\varepsilon} \quad \mathrm{Var}(\tilde{\varepsilon}) = E\left[\left(L^{-1} \otimes I_n\right)\varepsilon\varepsilon'\left(\left(L^{-1}\right)' \otimes I_n\right)\right] = I_m \otimes I_n$$

$$\tilde{y} = \left(L^{-1} \otimes I_n\right)y \quad \tilde{X} = \left(L^{-1} \otimes I_n\right)X$$

A normal prior for β, $\beta \sim N\left(\bar{\beta}, A^{-1}\right)$, is conjugate with the conditional likelihood for the transformed system. This means that we can apply the results from 2.8 and the posterior of β given Σ is normal.

$$\beta \mid \Sigma, y, X \sim N\left(\tilde{\beta}, \left(\tilde{X}'\tilde{X} + A\right)^{-1}\right) \quad \tilde{\beta} = \left(\tilde{X}'\tilde{X} + A\right)^{-1}\left(\tilde{X}'\tilde{y} + A\bar{\beta}\right) \tag{3.5.4}$$

As A gets small, the prior becomes flat and we recognize that the mean of this distribution is the Generalized Least Squares estimator.

The posterior of $\Sigma \mid \beta$ is in the IW form. To see this, first recognize that, given β, we "observe" or can compute the errors, ε. This means that, given β, the problem is the standard problem of inference regarding a covariance matrix using a multivariate normal sample. The IW prior is, therefore, conditional conjugate. If $\Sigma \sim IW\left(\nu_0, V_0\right)$, the posterior is in the form

$$\Sigma \mid \beta, y, X \sim IW\left(\nu_0 + n, S + V_{0,}\right) \quad S = E'E \quad E = \left[\varepsilon_1, \ldots, \varepsilon_m\right] \tag{3.5.5}$$

Again, if we let the prior precision go to zero, the posterior on Σ is centered over the sum of squared residuals matrix.

GIBBS SAMPLER FOR SUR MODEL[15]

Pick starting values, β_0, Σ_0 (note: Σ_0 must be a positive definite matrix)

Draw $\beta_1 \mid \Sigma_0$ from (3.5.4)[16]

[15] As in Chib and Greenberg (1995).

Draw $\Sigma_1 | \beta_1$ from (3.5.5)

Repeat

This sampler can be related to the non-Bayesian approach to estimating this model. Zellner originally proposed a feasible GLS procedure in which an estimate of Σ is formed by using residuals from equation by equation least squares estimates, $\hat{\Sigma} = \frac{1}{n}\hat{E}'\hat{E}$, where $\hat{E} = [e_1, \ldots, e_m]$ and $e_i = y_i - X_i\hat{\beta}_{LS,i}$. If we start the Gibbs sampler at this point and if we have a very diffuse prior on β, then the first iteration on β will be a draw from a distribution centered on the Zellner feasible GLS estimator. The Gibbs sampler takes this a step further and uses simulation to capture the uncertainty in both β and Σ. The finite sample distribution of the feasible GLS estimator is a nightmare due to the nonlinearities introduced by matrix inversion and multiplication. For this reason, econometricians have had to resort to asymptotic approximations. The sampling error in $\hat{\Sigma}$ does not figure in the asymptotic distribution of the "plug-in" or two-stage feasible GLS estimator. This shows the weakness of asymptotics. However, we no longer have to utilize these approximations. The Gibbs sampler for the SUR model performs extremely well with relatively trivial computation costs. Finally, note that if m=1, we have a sampler for the univariate regression model with a non-conjugate prior.

3.6 Conditional Distributions and Directed Graphs

One of the most common applications of the Gibbs Sampler is to hierarchical models. Hierarchical models are models constructed from a sequence of conditional distributions. More generally, we can construct a model by "connecting" or piecing together a set of

[16] Some care should be taken in the computations to draw β. The transformation of y and X involves very sparse matrices and can be optimized dramatically by taking advantage of the structure of these matrices.

conditional distributions in some sort of network or "graph." In this section, we provide a brief introduction to the basics of directed graphs. We will explain how to write a model as a directed graph and how to "read off" the Gibbs sampler from a graph.

The Bayesian paradigm starts with a prior and a likelihood. We can think of the prior as the first step and then we consider the distribution of the data given the model parameters. One way of remembering this "ordering" is to think about how we would simulate from the model (here model refers to the joint distribution of the unknowns and the data). First we would draw from the prior and then we would draw the data given the prior. This can be represented by a "directed acyclic" graph or DAG. A graph is a set of connected nodes. A directed graph has a notion of direction from node to node. An "acyclic" graph must have a direction from top to bottom with no "recirculation."

$$\begin{array}{ccc} p(\theta) & & p(y|\theta) \\ & & \\ \theta & \rightarrow & y \end{array} \tag{3.6.1}$$

A hierarchical model is specified through a sequence of two or more conditional distributions which specify the prior. This case of two conditional distributions can be represented as a directed graph as follows.

$$\begin{array}{ccccc} p(\theta_2) & & p(\theta_1|\theta_2) & & p(y|\theta_1) \\ \text{1st Stage} & & \text{2nd Stage} & & \\ & & & & \\ \theta_2 & \rightarrow & \theta_1 & \rightarrow & y \end{array} \tag{3.6.2}$$

Typically, θ_2 is of much lower dimension than θ_1. The sequence of two prior distributions can be thought of as a device to induce a marginal prior over θ_1.

$$p(\theta_1) = \int p(\theta_1, \theta_2) d\theta_2 = \int p(\theta_2) p(\theta_1|\theta_2) d\theta_2 \tag{3.6.3}$$

The hierarchical model in (3.6.2) specifies that θ_2 and y are independent conditional on θ_1 or that all dependence comes through θ_1. We can easily verify this by writing down the joint distribution.

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

There are two ways to see that this implies conditional independence. First, given θ_1, the joint distribution factors into two terms (represented by the functions f and g). Therefore, we have conditional independence. The other way to see this is to observe that there is no term involving all three variables, only a term (f) involving θ_1 and θ_2. This means that

$$p(\theta_2|\theta_1,y)\propto f(\theta_1,\theta_2) \quad \Rightarrow \; p(\theta_2|\theta_1,y)=p(\theta_2|\theta_1)$$

The hierarchical structure in (3.6.2) immediately suggests a "two-stage" Gibbs Sampler to simulate from the distribution of (θ_1,θ_2) given y.

(3.6.4)
$$\begin{gathered}\theta_2|\theta_1\\ \theta_1|\theta_2,y\end{gathered}$$

It is easily possible to write down more complicated directed graphs. However, some simple rules can help understand the structure of dependence implied by the graph. There are three sorts of local node arrangements.

The first is a linear set of three nodes

(3.6.5)
$$\theta_1\to\theta_2\to\theta_3$$

We have already seen an example of this in (3.6.2) (except that we do not "draw" y but condition on it). This structure has the basic conditional independence in it. A Gibbs sampler for (3.6.5) is given by

$$\begin{array}{c} \theta_1 | \theta_2 \\ \theta_2 | \theta_1 , \theta_3 \\ \theta_3 | \theta_2 \end{array} \tag{3.6.6}$$

Here $\theta_1, \theta_3 \perp | \theta_2$

The next structure looks different but has the same feature of conditional independence.

(3.6.7)

$$\begin{tikzcd} & \theta_2 \\ \theta_1 \arrow[ur] \arrow[dr] & \\ & \theta_3 \end{tikzcd}$$

The joint distribution implied by the graph in (3.6.7) is

$$p(\theta_1, \theta_2, \theta_3) = p(\theta_1) p(\theta_2 | \theta_1) p(\theta_3 | \theta_1)$$

Again, we have $\theta_2, \theta_3 \perp | \theta_1$.

However, the structure formed from two nodes pointing into one node does not display conditional independence.

(3.6.8)

$$\begin{tikzcd} \theta_1 \arrow[dr] & \\ & \theta_3 \\ \theta_2 \arrow[ur] & \end{tikzcd}$$

Here we have full dependence among all three random variables. The joint for (3.6.8) would be written $p(\theta_1, \theta_2, \theta_3) = p(\theta_1) p(\theta_2) p(\theta_3 | \theta_1, \theta_2)$. The Gibbs Sampler requires the full set of complete conditionals.

$$\begin{array}{c} \theta_1 | \theta_3, \theta_2 \\ \theta_3 | \theta_1, \theta_2 \\ \theta_2 | \theta_3, \theta_1 \end{array} \tag{3.6.9}$$

It is obvious that the "middle" conditional in (3.6.9) belongs in the sampler. What is less obvious is that the "top" and the "bottom" conditionals depend on a node that is more than one node away. But inspection of the joint distribution that the graph represents indicates that there is no conditional independence at all as there is one term in the joint involving all three parameters.

All directed graphs are made up of some combination of three examples above. This suggests three rules for "reading" the dependence structure from a graph:

A node depends on:

i. any node it points to
ii. any node that points to it
iii. any node that points to the node directly "downstream"

For example, consider the following graph.

(3.6.10)
$$\begin{tikzcd}
& \theta_2 \arrow[dr] & & \\
& & \theta_4 \arrow[r] & \theta_5 \\
\theta_1 \arrow[r] & \theta_3 \arrow[ur] & &
\end{tikzcd}$$

The Gibbs sampler for the graph in (3.6.10) is

(3.6.11)
$$\begin{gathered}
\theta_1 | \theta_3 \\
\theta_2 | \theta_4, \theta_3 \\
\theta_3 | \theta_1, \theta_4, \theta_2 \\
\theta_4 | \theta_2, \theta_3, \theta_5
\end{gathered}$$

3.7 Hierarchical Linear Models

In 3.5, we considered systems of regressions that are related through correlated errors. An alternative approach would be to relate regression equations through correlations in the regression coefficient vectors. This amounts to specifying a prior structure. Consider the set of regression equations.

(3.7.1)
$$y_i = X_i\beta_i + \varepsilon_i \qquad \varepsilon_i \sim \text{iid}N\left(0, \sigma_i^2 I_{n_i}\right) \quad i = 1, \ldots, m$$

We specify a different error variance for each equation but consider each regression to be independent of others. We tie together the equations by assuming that the $\{\beta_i\}$ have a common prior distribution.

$$\beta_i = \Delta' z_i + v_i \quad v_i \sim \text{iid}N\left(0, V_\beta\right) \tag{3.7.2}$$

(3.7.2) specifies a normal prior with mean $\Delta' z_i$ for each β. The d variables in the z vector represent characteristics of each of the m "cross-sectional" units or regression equations. A special case has $z_i = 1$ and $\Delta = \mu'$ which would have a common mean vector for all betas. This prior can be written as a Multivariate Regression Model.

$$B = Z\Delta + V \quad B = \begin{bmatrix} \beta_1' \\ \vdots \\ \beta_m' \end{bmatrix} \quad Z = \begin{bmatrix} z_1' \\ \vdots \\ z_m' \end{bmatrix} \quad \Delta = \begin{bmatrix} \delta_1 & \cdots & \delta_k \end{bmatrix} \quad v_i' \sim N\left(0, V_\beta\right) \tag{3.7.3}$$

B is m x k, Z is m x number of z variables (n_z), Δ is n_z x k. Each column of Δ has coefficients which describe how the mean of the k regression coefficients varies as a function of the variables in z. We also need a prior on the regression error variances. It is convenient to take a prior which specifies that each of the error variances is independent.

$$\sigma_i^2 \sim {}^{\nu_i s_{0,i}^2}\!\Big/_{\chi_{\nu_i}^2} \tag{3.7.4}$$

The prior in (3.7.3) specifies a fixed Δ matrix which determines the mean of the β distribution and a fixed V_β matrix which specifies the variance. Assessment of these priors could be difficult. Early "empirical Bayes" approaches simply estimate these parameters and then, conditional on these estimates, perform an approximate Bayesian analysis of each regression. A Full Bayesian solution can be obtained by specifying a further "stage" of priors on Δ and V_β.

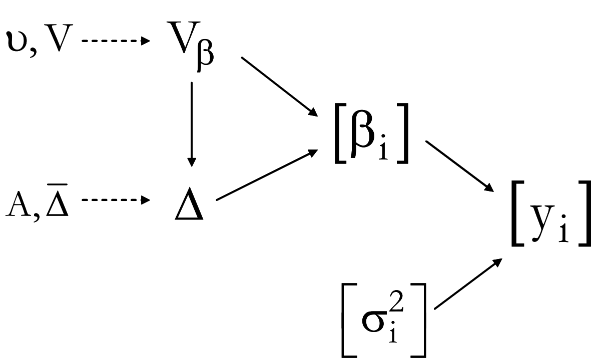
$$\begin{aligned} & V_\beta \sim \mathrm{IW}(\nu, V) \\ & \mathrm{vec}(\Delta) \big| V_\beta \sim N\left(\mathrm{vec}(\bar{\Delta}), V_\beta \otimes A^{-1}\right) \end{aligned} \tag{3.7.5}$$

The priors in (3.7.5) are the natural conjugate priors for the Multivariate regression model.

The prior on the collection of βs is specified through a two-stage process. First, we specify a normal prior on β and then a second stage prior on the parameters of this distribution. We can write out this model as a sequence of conditional distributions.

$$\begin{aligned} & y_i \big| X_i, \beta_i, \sigma_i^2 \\ & \beta_i \big| z_i, \Delta, V_\beta \\ & \sigma_i^2 \big| \nu_i, s_{0,i}^2 \\ & V_\beta \big| \nu, V \\ & \Delta \big| V_\beta, \bar{\Delta}, A \end{aligned} \tag{3.7.6}$$

The model above can also be written as a directed graph. The rules of directed graphs given in section 3.6 can be used to write down the Gibbs sampler for this model. In particular, a key observation is that all dependence between V_β, Δ and the data comes through the regression coefficients, $\{\beta_i\}$. The directed graph is shown below.



(3.7.6) converts the problem of assessing a prior on the m x k dimensional joint distribution of the βs into the problem of assessing hyper-parameters, $\bar{\Delta}, A, \nu, V$. A combination of the data and these hyper-parameters (plus the functional forms of the distributions) will influence the posterior on Δ and V_β. The Bayes estimators of the $\{\beta_i\}$ are of a shrinkage variety and will exhibit less variation than least squares estimates computed equation by equation. The amount of shrinkage will be dictated both by the prior hyperparameters and the data. If we assess a tight prior on a "small" covariance matrix by setting ν large and V to a small location value, then there will be a great deal of shrinkage. In addition, if the data suggests little variation in the β vectors from equation to equation, then the Bayes estimator will "adapt" to a posterior centered around a small value of V_β and the Bayes estimates of each beta vector will be shrunk even if our prior is relatively uninformative. As discussed in chapter 5, there are many subtle aspects of this normal hierarchical model. We will defer a full discussion of the prior used in this model until chapter 5.

Carl Morris was the first to observe that the Gibbs samplers were ideal for analysis of hierarchical models. The key observation is that given Δ and V_β, the $\left\{\beta_i, \sigma_i^2\right\}$ are independent with a prior which is the product of a normal prior on β_i and the inverse of a χ^2 prior. This can be analyzed via a Gibbs sampler by drawing $\beta_i \left| \sigma_i^2\right.$ and then $\sigma_i^2 | \beta_i$. Once the $\{\beta_i\}$ are drawn, they are sufficient for V_β and Δ. Given $\{\beta_i\}$, V_β and Δ can be drawn using the algorithm for the Multivariate regression model given in chapter 2, appendix 1. Thus, a Gibbs sampler for this model can be constructed by first drawing the regression parameters, $\left\{\beta_i, \sigma_i^2\right\}$, given the parameters of the first stage prior, Δ, V_β and then drawing the prior parameters conditional on $\left\{\beta_i, \sigma_i^2\right\}$. In the definition below, we will use τ_i to denote σ_i^2 to reduce notational clutter.

GIBBS SAMPLER FOR HIERARCHICAL LINEAR MODEL

Start with $\left\{\tau_i^0\right\}$, Δ^0, V_β^0

Draw

$$\beta_i^1 \left| y_i, X_i, \left(\Delta^0\right)^t z_i, V_\beta^0, \tau_i^0 \right.$$

and

$$\tau_i^1 \left| y_i, X_i, \beta_i^1, \upsilon_i, s_{0,i}^2 \right. \quad i = 1, \ldots, m$$

Draw

$$V_\beta^1 \left| \left\{\beta_i^1\right\}, \nu, V, Z, \bar{\Delta}, A \right.$$

and

$$\Delta^1 \left| \left\{\beta_i\right\}, V_\beta^1, Z, A, \bar{\Delta} \right.$$

Repeat as necessary

The draw of $\left\{\beta_i, \sigma_i^2\right\}$is conducted with a prior that specifies that β_i and σ_i^2 are a priori independent rather than dependent as in the natural conjugate prior, $\beta_i \sim N\left(\left(z_i \Delta\right)', V_\beta\right)$ and $\sigma_i^2 = {}^{\nu_i s_{0,i}^2} \big/ {}_{\chi_{\nu_i}^2}$.

$$\begin{aligned} &\beta_i^1 \left| y_i, X_i, z_i \Delta_0, V_\beta^0, \tau_i^0 \right. \sim N\left(\tilde{\beta}, \left(\tilde{X}_i' \tilde{X}_i + \left(V_\beta^0\right)^{-1}\right)^{-1}\right) \\ &\tilde{\beta} = \left(\tilde{X}_i' \tilde{X}_i + \left(V_\beta^0\right)^{-1}\right)^{-1} \left(\tilde{X}_i' \tilde{y}_i + \left(V_\beta^0\right)^{-1} \left(z_i' \Delta^0\right)'\right) \end{aligned} \tag{3.7.7}$$

$\tilde{y}_i = {}^{y_i} \big/ {}_{\sigma_i}$ and $\tilde{X}_i = {}^{X_i} \big/ {}_{\sigma_i}$.

The R code for this sampler is in `rhierLinearModel` which is available in our package, *`bayesm`*. This code makes use of the function to draw from the posterior of a Multivariate Regression model given at the end of chapter 2. Even though the Gibbs sampler requires looping, the bulk (about 80 per cent) of the computing time is devoted to drawing the regression coefficient vectors. For 88 regressions with about 60 observations each and three independent variables, this code allows for about 1500 draws per minute.

To illustrate the functioning of this sampler, consider a very typical problem in promotional response modeling. Most Consumer Packaged Goods (CPG) manufacturers have data on the pricing and promotional activities of many of their "key" accounts. A "key" account is a combination of a retailer and market area. For example, Safeway-Denver is a key account that receives attention from P&G's salesforce. The data is usually weekly data for one to two years. P&G may define several hundred key accounts. In order to allocate funds and saleforce effort over these accounts, it would be useful to understand how the customers in these accounts respond to various types of promotional activity. We can use data on sales and measure of price and promotional activity to estimate a simple sales response model for each of these accounts.

As an example, consider the sliced cheese product manufactured by Borden. This dataset can be loaded from *`bayesm`* using the command `data(cheese)`. Data is available on some 88 key accounts for an average of 65 weeks. Weekly observations are recorded of unit volume (number of units sold in all stores in this account during each week), price in $, and a measure of display activity. Displays are a form of in-store advertising that usually consists of special signage or display of the merchandise in a prominent location. The measure of display activity is a "percent of ACV on display." Kraft would like to see retailers use the optimal combination of display and pricing to promote and sell this product.

One straightforward approach would be to run 88 separate regressions each with about 60 observations. However, the independent variables do not always have very much variation at the account variable. For example, two of the accounts have no display activity in this period. Even ignoring the fact that the display coefficient is not estimable for two of the 88 regressions, the least squares coefficients are not very usable as shown in figure 3.6. These are coefficients from a regression of ln(Volume) on Display and ln(Price). Some of the display coefficients are absurdly large. To interpret the coefficient, recall that the Display variable reaches a maximum of 1 which means 100 per cent display coverage. Thus, a display coefficient of 5 implies a multiple of sales volume of e^5 or almost a 150 fold increase. Clearly, these coefficients have been influenced by sampling error and, perhaps, outlying observations.

Figure 3.6 plots the least squares coefficients against the posterior means obtained from the Linear Hierarchical model Gibbs sampler[17]. The prior settings are

$$\nu_i = 3,\ s_{0,i}^2 = \mathrm{Var}\left(y_i\right), \nu = k + 3, V = \nu \times .1I_k,\ \bar{\Delta} = 0,\ A = .01$$

These prior settings represent a proper but very diffuse prior. Even so, the posterior means display a strong shrinkage effect. The absurdly large values of the least squares coefficients are shrunk in towards more reasonable values. This shrinkage stems from two forces: 1). the adaptive nature of the Hierarchical model which adapts V_β to the observed variation in the data and 2). the normal first stage prior which has very thin tails. It should be noted that, for most accounts, the least squares and posterior means are similar. This means that the sample information dominated the prior for these accounts (note that the prior is centered over 0 for the mean of the regression coefficients). The prior is set to a very

[17] Here there are no "z" variables, so Z is simply a vector of ones and Δ is a common mean vector for the coefficients.

diffuse setting so this is not very surprising. There are several coefficients for which both the posterior mean and the least squares estimates are negative. A negative coefficieint would imply the displays depressed sales. This might be something that one would want to rule out a priori. Unfortunately, the conditional conjugate priors employed in this model are not capable of imposing a sign restriction on the regression coefficients. We will consider this in chapter 5.

To illustrate how the prior settings affect the degree of shrinkage, we compute posterior means for three different values of $\nu, k+3, k+.5\bar{n}, k+2\bar{n}$. $\bar{n}$ is the average number of observations in each data set. The three values of ν tighten down the IW prior on V_β. The V location matrix has already been set to a small value. This means that if ν is large, our prior on V_β is highly informative and located over small values, inducing a great deal of shrinkage. Figure 3.7 plots the least squares coefficients vs. the posterior means for three levels of ν and for each of the three coefficients in the model. The small value of ν is blue, the medium green and the high value yellow. The shrinkage is rather dramatic for ν representing prior information obtained from a sample roughly twice the size of each regression sample. Although the IW and Wishart distributions only have one tightness parameter, this does not imply that the degree of shrinkage is the same for each coefficient. The degree of shrinkage depends on the amount of sample information available for each coefficient. The display coefficient is most difficult to estimate as displays are relatively rare compared to price changes. The intercept is the easiest to estimate and displays the least shrinkage.

3.8 Data Augmentation and a Probit Example

The examples that we have seen so far show that the Gibbs Sampler is extremely well-suited sets of linear models for which the conditional distributions are known and for which standard methods can be used to make direct draws from these conditional distributions. We enlarge the set of models that can be analyzed by requiring only conditional and not full conjugacy. This allows for analysis of systems of linear models which, heretofore, required approximate methods. However, the Gibbs Sampler can be applied to a much wider class of models, once the principal of data augmentation is introduced. The idea of data augmentation has its origin in the literature on missing values and the EM algorithm. The idea that missing values are unobserved and, therefore, should properly be considered as part of the "parameter" vector comes naturally to a Bayesian. In the EM algorithm for missing data models, the missing data is replaced with its expectation conditional on the observed data and the "complete" data likelihood is maximized over the "parameters." To a Bayesian, it is much more natural to compute the joint posterior of the missing values and the "parameters" and simply margin down to the parameters if this is all that is of interest. The idea of data augmentation extends to any situation in which there are unobservable constructs. For example, many distributions can be written as mixtures of other distributions so that data augmentation can be used to form a Gibbs sampler for these problems. Of particular relevance for marketing problems, is the use of latent variables to formulate models with discrete lumps of probability.

To illustrate the usefulness of the data augmentation concept for discrete dependent variables models, consider the latent variable formulation of the binary Probit model.

(3.8.1)
$$z_i = x_i^{'}\beta + \varepsilon_i \quad \varepsilon_i \sim N(0,1)$$
$$y_i = \begin{cases} 0 \text{ if } z_i < 0 \\ 1 \text{ otherwise} \end{cases}$$

We observe (X, y). If this model is used to represent the choice between two alternatives, then z has the interpretation as the difference in utility between the two alternatives. We choose alternative "A" if it is more attractive than "B." We only partially observe latent utility in the sense that only x is observed. Other influences on utility are represented by the "error" term. Given that the latent structure in (3.8.1) is a standard regression model, we can use a normal prior for $\beta, \beta \sim N\left(\bar{\beta}, A^{-1}\right)$.

Data augmentation proceeds by considering the entire vector of n z values as part of the parameter vector, $\theta^{'} = (z, \beta)$. Given the normal prior for β, the model is complete in the sense that (3.8.1) specifies the joint distribution[18] of z and β.

$$p(z, \beta | X) = p(z | \beta, X)\, p(\beta)$$

We note that this is a highly correlated distribution, particularly if the prior on β is diffuse. We can write the directed graph for this model in (3.8.2).

(3.8.2)
$$\beta \rightarrow \quad z \rightarrow \quad y$$

This directed graph immediately reveals that β, y are independent conditional on z.

The posterior distribution of θ can easily be computed by using a Gibbs sampler[19].

(3.8.3)
$$z | \beta, X, y$$

(3.8.4)
$$\beta | z, X$$

[18] If you regard the z values as meaningful objects for inference, then this is a prior distribution. If not, then the augmented parameters are simply devices by which one gets at the posterior distribution of β.

This Gibbs sampler recognizes that θ separates into two natural groups or "blocks." Given β and the data, the z's are independent truncated univariate normal distributions. Given z, inference on β is just a Bayes linear regression analysis with a normal prior and no scale parameter (note that z is sufficient for β and we don't need to add y to the conditioning arguments (3.8.4)).

GIBBS SAMPLER FOR BINARY PROBIT

```
    Start with $\beta_0$

    Draw z from (3.8.3) by making n independent draws from $TN_{(a_i,b_i)}\left(-x_i'\beta_0,1\right)$
        $a_i = 0$ if $y_i = 1, -\infty$ otherwise; $b_i = 0$ if $y_i = 0$, $\infty$ otherwise

    Draw $\beta_1$ from (3.8.4) using standard normal theory
        $\beta|y,X \sim N\left(\tilde{\beta},\left(X'X+A\right)^{-1}\right)$; $\tilde{\beta} = \left(X'X+A\right)^{-1}\left(X'y+A\bar{\beta}\right)$

    Repeat as necessary
```

The dependence between draws of β comes entirely through the z vector. This will be useful Gibbs sampler to the extent to which the latent variables and β are not too highly correlated. R code for this binary probit sampler is available as the function, `rbprobitGibbs`, in our R package, *bayesm*.

Figure 3.8 shows the results of running this Gibbs Sampler with a simulated dataset with n=100, two regressors which are uniform on (0,1) and independent. The left hand side shows histograms of the marginal posterior distribution of each parameter. The vectical line marks the "true" value of the parameter underlying the simulated data. The right hand panels show the autocorrelation functions. The marginal posterior distributions of the

[19] Albert and Chib (1993) were the first to propose this sampler (see also, Chib (1992), for a closely related

model parameters are very normal and the ACFs for this data are very reasonable even though there is not much information in this data on the betas.

One might be tempted to conclude that all of this machinery is an elaborate way of producing results from asymptotic theory (although one would have to verify that the posterior covariance is close to the asymptotic covariance). However, the parameters of this model are not as directly interpretable and relevant as in the linear regression context. In marketing applications, we are often interested in predicted probabilities for given values of x. Figure 3.9 shows the posterior distribution of the probability y=1 for various x vectors, i.e. the posterior distribution of $\Phi\left(x'\beta\right)$. Since probabilities are bounded and there is posterior uncertainty, these distributions are very non-normal. Superimposed on the histogram are normal densities evaluated at the posterior mean and variance.[20] Even this small problem and highly regular model provide a powerful motivation to eliminate asymptotic approximations.

Finally, we should note that this model can easily be investigated by importance sampling performed on the posterior obtained by integrating out z.

$$p\left(\beta\middle|X,y\right)\propto p\left(\beta\right)\prod_i \Phi\left(x_i'\beta\right)^{y_i}\left(1-\Phi\left(x_i'\beta\right)\right)^{1-y_i} \tag{3.8.5}$$

In fact, per unit of computing time, a direct approach via importance sampling will surely yield more information than the binary Gibbs Sampler due to the autocorrelation in the draw sequence. The binary probit Gibbs Sampler is of interest primarily because it suggests that data augmentation strategies can be useful for the multinomial probit and other problems for which the likelihood over the model parameters is difficult to evaluate.

model).

[20] Asymptotic theory would be even worse since it would keep the normal distribution assumption but insert asymptotic estimates of the mean and variance.

3.9 Mixtures of Normals

Finite mixtures of multivariate normal distributions can provide a very flexible model for multivariate data. Mixtures of normals can accommodate thick tailed and skewed distributions. However, in the multivariate case, the possibilities are even broader. For example, we can create a joint distribution with "banana" shaped contours by arranging closely spaced normal distributions along a curve. There is a sense that with enough mixture components, one can approximate any multivariate distribution in the same sense that you can build any shaped hill by piling up small mounds of gravel.

The basic mixture of normals model can be written

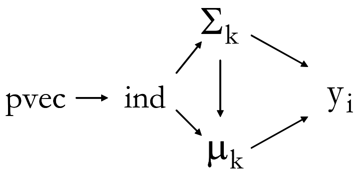
$$\begin{gathered} y_i \sim N\left(\mu_{\text{ind}_i}, \Sigma_{\text{ind}_i}\right) \\ \text{ind}_i \sim \text{multinomial}\left(\text{pvec}\right) \end{gathered} \tag{3.9.1}$$

Here y_i is a p-dimensional vector and pvec is a vector of K mixture probabilities. This model is referred to as a mixture of normals with K components. (3.9.1) is a direct model for simulation from a mixture of normals (see `rmixture` in our R package, ***bayesm***). First, we draw a multinomial distributed indicator of which component is "active" and then we draw a multivariate normal vector from this component. This representation of the model also suggests the basis of a Gibbs Sampler by augmenting the parameters with the vector of n indicators.

Priors for the mixture of normals model can be taken in convenient conditionally conjugate forms

$$\begin{aligned} &\text{pvec} \sim \text{Dirichlet}\left(\alpha\right) \\ &\mu_j \sim N\left(\bar{\mu}, \Sigma_k \otimes a_\mu^{-1}\right) \\ &\Sigma_j \sim IW\left(\upsilon, V\right) \\ &k = 1, \ldots, K \end{aligned} \tag{3.9.2}$$

In (3.9.2), the joint prior on the normal component parameters is independent conditional on p and in the form of the natural conjugate prior for multivariate regression (see section 2.8.5). The DAG for the mixture of normals model can be written as



Given this DAG, we can easily write down the Gibbs Sampler as consisting of the following sets of conditionals.

$$\begin{aligned} &\text{ind}\,|\,\text{pvec},\{\mu_k,\Sigma_k\},Y \\ &\text{pvec}\,|\,\text{ind} \\ &\{\mu_k,\Sigma_k\}\,|\,\text{ind},Y \\ &k=1,\ldots,K \end{aligned} \qquad (3.9.3)$$

Y is the n x p matrix of multivariate observations. This sampler was introduced by Diebolt and Robert (1994). The key idea is that once the indicators are drawn, the observations are classified by normal component and then one can proceed with K independent conjugate draws of the normal component parameters.

The draw of the indicators is a multinomial draw based on the likelihood ratios with p as the prior probability of membership in each component.

$$\begin{aligned} &\text{ind}_i \sim \text{multinomial}\left(\pi_i\right);\ \pi' = \left(\pi_{i,1},\ldots,\pi_{i,K}\right) \\ &\pi_{i,k} = \text{pvec}_k \frac{\varphi\left(y_i\,|\mu_k,\Sigma_k\right)}{\Sigma_k\,\varphi\left(y_i\,|\mu_k,\Sigma_k\right)} \end{aligned} \qquad (3.9.4)$$

Here $\varphi(\bullet)$ is the multivariate normal density.

The draw of pvec given the indicators is a Dirichlet draw

$$\begin{aligned}&\text{pvec} \sim \text{Dirichlet}\left(\tilde{\alpha}\right)\\ &\tilde{\alpha}_k = n_k + \alpha_k \\ &n_k = \sum_{i=1}^{n} I\left(\text{ind}_i = k\right)\end{aligned} \qquad (3.9.5)$$

The draw of each $\left(\mu_k, \Sigma_k\right)$ can be made using the algorithm to draw from the Multivariate regression model as detailed in section 2.8.5. For each subgroup of observations, we have an MRM model of the form

$$Y_k = \iota\mu_k^{'} + U; \;\; U = \begin{bmatrix} u_1^{'} \\ \vdots \\ u_{n_k}^{'} \end{bmatrix}; \; u_i \sim N\left(0, \Sigma_k\right) \qquad (3.9.6)$$

Here Y_j is the submatrix of Y that consists of the n_j rows where $\text{ind}_i = j$. The results of chapter 2 simplify to the following draws:

$$\begin{aligned}&\Sigma_k \left| \Theta_k^*, \upsilon, V \sim IW\left(\upsilon + n_k, V + S\right)\right.\\ &\mu_k \left| \Theta_k^*, \Sigma_k, \bar{\mu}, a_\mu \sim N\left(\tilde{\mu}_k, {}^{1}\!\!\diagup_{\left(n_k + a_\mu\right)} \Sigma_k\right)\right.\end{aligned} \qquad (3.9.7)$$

where

$$\begin{aligned}&S = \left(\Theta_k^* - \iota\tilde{\mu}_k^{'}\right)^{'}\left(\Theta_k^* - \iota\tilde{\mu}_k^{'}\right)\\ &\tilde{\mu}_k = \left(n_k + a_\mu\right)^{-1}\left(n_k\bar{\theta}_k^* + a_\mu\bar{\mu}\right)\\ &\bar{\theta}_k^* = \left(\Theta_k^{*'}\iota / n_k\right)^{'}\end{aligned} \qquad (3.9.8)$$

This Gibbs sampler for mixtures of normals is available as function, `rnmixGibbs`, in *`bayesm`*.

3.9.1 Identification in Normal Mixtures

It is well known that the likelihood for the normal mixture model can have up to K! symmetric modes. This is due to what is referred to as the label-switching "problem." We can simply interchange or permute the labels for each of the components and have the same value of the likelihood. For example, consider the mixture of two univariate normal distributions. There are two equal height posterior modes (assuming the priors are identical). We can simply interchange the labels, calling mode 1 "2" and mode 2 "1" and leave the likelihood unchanged. This means that the marginal posteriors of the mean parameters will often have two modes. As we know from the examples considered in chapter 3, this can affect many MCMC algorithms. It is possible that the algorithm may only investigate one of the modes or some subset of modes, leaving others completely untouched. This will occur when there is strong separation or classification information in the "data." However, when differences between components are small, there can be a good deal of switching from mode to mode. This switching from mode to mode by MCMC methods is often what researchers in this area mean by "label-switching."

Figure 3.10 illustrates the label-switching problem by considering the problem of inference about a mixture of two normals. Component "1" is a N(1,1) random variable and component "2" is ~N(2,1). The mixture probability is .5 . This is an example where there is little distance between the modes for each component. The mixture is a symmetric and unimodal density centered at 1.5. If we have a modest number of observations, the Gibbs Sampler which "flip" the component labels so that what had been labeled a draw of the mean for the first component is now the mean of the second component. $\left(\mu_1 \leftrightarrow \mu_2\right)$. Figure 3.10 plots the Gibbs Sampler output for μ_1 (labeled "1") and μ_2 (labeled "2"). We can see several label switches, including one around draw number 50.

This means that we cannot simply look at the marginal distribution of μ_1 from our Gibbs Sampler output. This parameter is not identified. However, the joint density is identified.

(3.9.9) $$p(y) = p\varphi(y|\mu_1,\sigma_1) + (1-p)\varphi(y|\mu_2,\sigma_2)$$

For any given value of y, we can compute the posterior distribution of the density at this value. If we use a grid of y values, we can compute the posterior distribution of the entire density of the data. In particular, we can average the posterior to obtain the posterior mean as an estimator of the density. Figure 3.11 shows ten posterior draws of the fitted density in (3.9.9). The solid density is the true density.

Much of the early work with mixture models attempted to solve the identification problem for individual component parameters by using various a priori ordering restrictions. For example, some advocate ordering the components by prior probability.

(3.9.10) $$p_1 > p_2 > \dots p_K$$

Unfortunately, (c.f. Stevens 2000) imposing this restriction does not necessarily remove the identification problem. Other proposals include ordering by the normal mixture component parameters. Obviously, it may be difficult to define an ordering in the case of mixtures of *multivariate* normal distributions. Moreover, even in the cases of univariate normals, Stevens (2000) has noted that identification cannot always be achieved. Choosing the right way to divide the parameter space so as to insure that only one mode remains can be somewhat of an art form that may be close to impossible in high dimensional problems (see Fruhwrith-Schnatter (2001) for a suggested method that may work in low dimensional situations).

Instead of imposing what can been termed "artificial" identification constraints, Stevens (2000) advocates postprocessing via relabelling of MCMC draws so as to minimize some sort of statistical criterion such as divergence of the estimated marginal posteriors from some unimodal distribution. This means that post-simulation optimization methods must be used to achieve a relabelling in the spirit of clustering algorithms. There is no guarantee that such post processing will necessarily uncover the true structure of the data and there is still “art” in the choice of objective function and tuning of the re-labelling.

The label-switching phenomenon is only a problem to the extent to which the investigator wishes to attach meaning to specific normal components. If one interprets each normal component as a sub-population, then we might want to make inferences about the means and covariances for that component. For example, we might think of the population of respondents to a questionnaire as comprised of distinct groups each with some heterogeneity within group but where this may be small relative to across group heterogeneity. In this situation, label-switching becomes an issue of identification for the parameters of interest. The only way to “solve” this identification problem is by use of prior information. However, if we regard the mixture of normals as a flexible approximation to some unknown joint distribution, then the label-switching “problem” is not relevant. Ultimately, the data identify the joint distribution and we don’t attach substantive importance to the mixture component parameters. We should recognize that this runs somewhat counter to the deeply ingrained tradition of identifying consumer segments. We do not really think of segments of homogeneous consumers as a reality but merely a convenient abstraction for the purpose of marketing strategy discussions. The empirical evidence, to date, overwhelmingly favors the view that there is a continuum of consumer tastes.

Thus, we view the object of interest as the joint density of the parameters. This density and any possible functions defined on it such as moments or probabilities of regions are identified without a priori restrictions or ad hoc post-processing of MCMC draws. Once you adopt this point of view, the identification and label-switching literature becomes irrelevant and, instead, you are faced with the problem of summarizing a fitted multivariate density function. Lower dimensional summaries of the mixture of normals density are required (as Fruthwrith-Schnatter et al (2004) point out any function of this density will also be immune to the label-switching identification issues). One possible summary would be the univariate marginal densities for each component of θ. However, this does not capture co-movement of different elements. It is certainly possible to compute posterior distributions of covariance but, as pointed out above, these lose interpretability for multi-modal and non-normal distributions.

3.9.2 Performance of the Unconstrained Gibbs Sampler

The Gibbs sampler outlined in (3.9.3) is referred to as the "unconstrained" Gibbs sampler in the sense that no prior constraints have been imposed to achieve identification of the mixture component parameters. As such, this sampler may exhibit label-switching. As pointed out above, any function which is invariant to label-switching such as the estimated mixture density will not be affected by this problem. There also may be algorithmic advantages to not imposing identification constraints. As pointed out by Fruhwirth-Schatter (2001), identification constraints hamper mixing in single-move constrained Gibbs samplers. For example, if we imposed the constraint that the mixture probabilities must be ordered, then we must draw from a prior distribution restricted to a portion of the parameter space in which the ordering is imposed. A standard way to do this is to draw each of the K probabilities, one by one, given the others. The ordering constrains mean that as we draw the kth probability, it must lie between the k-1st and k+1st probabilities. This may leave little room for navigation. Thus, the unconstrained Gibbs sampler will often mix better than constrained samplers. Fruhwirth-Schnatter introduces as "random permutation" sample to promote better navigation of all modes in the unidentified parameters space of the mixture component parameters. This may improve mixing in the unidentified space but will not improve mixing for the identified quantities such as the estimated density and associated functions. The fact that the unconstrained Gibbs sampler may not navigate all or even more than one of the K! symmetric modes does not mean that it doesn't mix well in the identified space. As Gilks (1997) comments "I am not convinced by the … desire to produce a unique labeling of the groups. It is unnecessary for valid Bayesian inference concerning identifiable quantities; it worsens mixing in the MCMC algorithm; it is difficult to achieve in any meaningful way, especially in high dimension, and it is therefore of dubious explanatory or predictive value" (p. 771).

Our experience is that the unconstrained Gibbs Sampler works very well, even for multivariate data and with a large number of components. To illustrate this, consider 5 dimensional data simulated from a 3 component normal mixture, n=500.

(3.9.11)
$$\mu_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}; \ \mu_2 = 2\mu_1; \ \mu_3 = 3\mu_1; \ \ \Sigma_k = \begin{bmatrix} 1 & .5 & \cdots & .5 \\ .5 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & .5 \\ .5 & \cdots & .5 & 1 \end{bmatrix}$$
$$\text{pvec} = \begin{pmatrix} 1/2 \\ 1/3 \\ 1/6 \end{pmatrix}$$

We started the unconstrained Gibbs sampler with 9 normal components. Figure 3.12 represents the distribution of the indicator variable across the 9 components for each of the 400 first draws. The width of each horizontal line is proportional to the frequency with which that component number occurs in the draw of the indicator variable. The sampler starts out with an initial value of the indicator vector which is split evenly among the nine components. The sampler quickly shuts downs a number of the components. By 320 or so draws, the sampler is visiting only three components with frequency corresponding to the mixture probabilities that were used to simulate the data.

Figure 3.13 shows each of the five marginal distributions for particular draws in the MCMC run shown in figure 3.13. The solid lines are the "true" marginal distributions implied by the normal mixture given in (3.9.11). The top panel shows the draw of the five marginal distributions for the 4th draw of the sampler. The marginals vary in the extent to which they show "separation" in modes. The marginal of the 5th component has the most pronounced separation. After only 4 draws, the sampler has not yet "burned-in" to capture the multimodality in the marginal distribution of the 3rd through 5th components. However, as shown in the bottom panel, by 100 draws the sampler has found the rough shape of the marginal distributions. Variation from draw to draw after the 100th draw simply reflects posterior uncertainty.

3.10 Metropolis Algorithms

The Gibbs sampler is enormously useful, particularly for models built up from hierarchies of relatively standard distributions. However, there are many problems for which the conditional distributions are not of a known form that is easy to simulate from. For this reason, it is useful to have a more "general purpose" tool. By "general purpose," we mean a tool that can be applied to, at least in principle, to any posterior distribution. Of course, in practice, there will be problems for which the general purpose algorithm will produce a poorly performing MCMC sampler. This means that the general purpose algorithm means an approach to generating a candidate MCMC method for virtually any problem. The performance of this candidate will not be assured and must still be investigated.

The Metropolis class of algorithms is a general purpose approach to producing Markov chain samplers.[21] The idea of the Metropolis approach is to generate a Markov Chain with the posterior, $\pi(\ )$, as its invariant distribution by appropriate modifications to a related Markov chain that is relatively easy to simulate from. Viewed in this way, Metropolis algorithms are similar in spirit to the accept/reject method of iid sampling. Accept/reject methods sample from a proposal distribution and then reject draws to modify the proposal distribution to achieve the desired target distribution. The proper invariant distribution is achieved by constructing a new chain that is time reversible with respect to π.

We will start with a discrete state space version of the Metropolis algorithm which will illustrate the essential workings of the algorithm. We have a transition matrix, Q, which we want to modify to insure that the resultant chain has a stationary distribution given by the vector π. The dimension of the state space is d.

DISCRETE METROPOLIS ALGORITHM

```
    Start in state i, $\theta_0 = \theta^i$
    Draw state j with probability given by $q_{ik}, k = 1,\ldots,d$ (multinomial draw)
    Compute $\alpha = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}$
    With prob $\alpha$, $\theta_1 = \theta^j$ (move) ELSE $\theta_1 = \theta^i$ (stay)
    Repeat, as necessary
```

[21] Although the Metropolis method was developed for discrete distributions and discussed in the statistics literature as early as 1970 (Hastings, 1970), the popularity of this method is due in large part to Tierney (1994) and Chib and Greenberg (1995) who provided a tutorial in the method as well as many useful suggestions for implementation.

Note that only the ratios, ${\pi_j}/{\pi_i}$, are required and we only need to know posterior distribution up to a constant.

The unique aspect of this algorithm is the possibility that it will not move on a given iteration of the chain. With probability $1-\alpha$, the chain will repeat the value from the rth to (r+1)st iteration. These repeats are to insure that new chain is reversible. If $\pi_i q_{ij} > \pi_j q_{ji}$, then there will be "too many" transitions from state i to state j and not enough reverse transitions from state j to i. For this reason, the Metropolis chain only accepts the α fraction of the transitions from i to j and all of the moves from j to i.

This algorithm is constructed to be time reversible with respect to π. To see this, recall that time reversibility requires $\pi_i p_{ij} = \pi_j p_{ji}$. The transition probability matrix for the chain defined by the Metropolis algorithm is defined by

$$p_{ij} = q_{ij}\alpha(i,j)$$

Therefore,

$$\pi_i p_{ij} = \pi_i q_{ij} \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\} = \min\left\{\pi_i q_{ij}, \pi_j q_{ji}\right\}$$

$$\pi_j p_{ji} = \pi_j q_{ji} \min\left\{1, \frac{\pi_i q_{ij}}{\pi_j q_{ji}}\right\} = \min\left\{\pi_j q_{ji}, \pi_i q_{ij}\right\}$$

and the condition for time reversibility is satisfied.

The continuous version of the Metropolis algorithm has exactly the same formulation as the discrete case except that the analysis of reversibility and convergence is slightly more complex. In particular, the transition kernel for the Metropolis does not have a standard density but rather has a mixture of discrete and continuous components. The continuous state space Metropolis algorithm starts with a proposal transition Kernel defined by the transition function $q(\theta,\vartheta)$. Given θ, $q(\theta,\bullet)$ is a density. The continuous state space version of the Metropolis algorithm is as follows.

CONTINUOUS STATE SPACE METROPOLIS

```
    Draw $\vartheta \sim q(\theta_0,\bullet)$
    Compute $\alpha(\theta,\vartheta) = \min\left\{1, \frac{\pi(\vartheta)q(\vartheta,\theta)}{\pi(\theta)q(\theta,\vartheta)}\right\}$
    With prob $\alpha$, $\theta_1 = \vartheta$ ELSE $\theta_1 = \theta_0$
    Repeat as necessary
```

To see that the continuous version of the Metropolis algorithm has π as its invariant distribution[22], we first define the Kernel. Recall that the Kernel provides the probability that the chain will advance to a set A given that it is currently at point θ.

$$K(\theta,A) = \int_A p(\theta,\vartheta)d\vartheta + r(\theta)\delta_A(\theta) \tag{3.10.1}$$

$$p(\theta,\vartheta) = \alpha(\theta,\vartheta)q(\theta,\vartheta) \tag{3.10.2}$$

$$\delta_A(\theta) = \begin{cases} 1 & \theta \in A \\ 0 & \text{otherwise} \end{cases}$$

p defined in (3.10.2) is the transition function for the Metropolis chain. The probability that the chain will move away from θ is given by

$$\int_\Theta p(\theta,\vartheta)d\vartheta = \int_\Theta \alpha(\theta,\vartheta)q(\theta,\vartheta)d\vartheta$$

Since $K(\theta,S)=1$, this implies that $r(\theta)$ is the probability that the chain will stay at θ. Since there is a possibility for the chain to repeat the value of θ and, therefore, given θ, we

[22] Here we follow the exposition of Chib and Greenberg (1995).

can't provide a standard density representation for the distribution of ϑ since there is a mass point at θ. The conditional distribution defined by the Metropolis kernel can be interpreted as a mixture of a mass point at θ and a continuous density $p(\theta,\vartheta) \big/ \left(1-r(\theta)\right)$.

The Metropolis transition function p satisfies the "detailed balance" condition

$$\pi(\theta)p(\theta,\vartheta)=\pi(\vartheta)p(\vartheta,\theta) \tag{3.10.3}$$

or $\pi(\theta)\alpha(\theta,\vartheta)q(\theta,\vartheta)=\pi(\vartheta)\alpha(\vartheta,\theta)q(\vartheta,\theta)$ which is true by the construction of the α function. (3.10.3) ensures that the Metropolis chain will be time reversible. We can now show that (3.10.3) implies that π is the invariant distribution of the chain. Recall that, for a continuous state chain, π is the invariant distribution if $\int_A \pi(\theta)d\theta=\int_\Theta \pi(\theta)K(\theta,A)d\theta$.

$$\int_\Theta \pi(\theta)K(\theta,A)d\theta=\int_\Theta \pi(\theta)\left[\int_A p(\theta,\vartheta)d\vartheta+r(\theta)\delta_A(\theta)\right]d\theta$$

$$=\int_\Theta\int_A \pi(\theta)p(\theta,\vartheta)d\vartheta d\theta+\int_\Theta \pi(\theta)r(\theta)\delta_A(\theta)d\theta$$

Interchanging the order of integration and applying the detailed balanced equation,

$$=\int_A\int_\Theta \pi(\vartheta)p(\vartheta,\theta)d\theta d\vartheta+\int_\Theta \pi(\theta)r(\theta)\delta_A(\theta)d\theta$$

$$=\int_A \pi(\vartheta)\left[\int_\Theta p(\vartheta,\theta)d\theta\right]d\vartheta+\int_A \pi(\theta)r(\theta)d\theta$$

The integral of the Metropolis function in the first term above is simply the probability of moving.

$$=\int_A \pi(\vartheta)\left[1-r(\vartheta)\right]d\vartheta+\int_A \pi(\theta)r(\theta)d\theta$$

$$=\int_A \pi(\vartheta)d\vartheta$$

Convergence of the Metropolis algorithm is assured by positivity of the proposal transition function $q(\theta, \vartheta)$[23] which assures that the chain is irreducible (see Robert and Casella, section 7.3.2). The challenge is to choose a "proposal" or candidate distribution that is relatively easy to evaluate and simulate from and yet produces a Metropolis chain with acceptable convergence properties. There are a wide variety of different styles of proposal densities. We will review some of the most useful.

3.9.1 Independence Metropolis Chains

Importance Sampling relies on having a reasonable approximation to π. Usually, the importance function is based on an asymptotic approximation to the posterior with fattened tails. This idea can be embedded in a Metropolis chain by taking q to be independent of the current value of the chain, $q(\theta, \vartheta) = q_{\text{imp}}(\vartheta)$ and based on the same sort of importance function ideas. We denote the independent Metropolis transition density by q_{imp} to draw the close analogy with importance sampling and with the criterion for a useful importance function.

INDEPENDENCE METROPOLIS

```
    Start with $\theta_0$
    Draw $\vartheta \sim q_{\text{imp}}$
    Compute $\alpha = \min\left\{1, \frac{\pi(\vartheta)q_{\text{imp}}(\theta)}{\pi(\theta)q_{\text{imp}}(\vartheta)}\right\}$
    With prob $\alpha$, $\theta_1 = \vartheta$, ELSE $\theta_1 = \theta_0$
    Repeat, as necessary
```

[23] An additional condition is required to insure that the Metropolis chain is aperiodic. This condition requires that there be a non-zero probability of repeating.

If q is an excellent approximation to π, then most draws will be accepted since the ratio in the α computation will be close to one. This means that we will have a chain with almost no autocorrelation. To understand how the chain handles discrepancies between q and π, rewrite the ratio in the α computation as

$$\frac{\pi(\vartheta)q_{\text{imp}}(\theta)}{\pi(\theta)q_{\text{imp}}(\vartheta)}=\frac{\pi(\vartheta)/q_{\text{imp}}(\vartheta)}{\pi(\theta)/q_{\text{imp}}(\theta)}.$$

If π has more relative mass at ϑ than at θ, the chain moves to ϑ with probability one to build up mass at ϑ. On the other hand, if π has less relative mass at ϑ than at θ, then there is a positive probability that the chain will repeat θ which builds up mass at that point and introduces dependence in the sequence of chain draws. This is really the opposite of accept/reject sampling in which the proposal distribution is "whittled down" to obtain π by rejecting draws.

It is important that the q proposal distribution has fatter tails than the target distribution for the same intuition as applies to importance sampling. If the target distribution has fatter tails than the proposal, the chain will wander off into the tails and then start repeating values to build up mass. If the proposal distribution dominates the target distribution in the sense that we can find a finite number M such that $\pi \le Mq_{\text{imp}}$ for all $\theta \in \Theta$, then the independence Metropolis chain is ergodic in an especially strong sense in that the distance between the repeated Kernel and π can be bounded for any starting point θ_0 (see Robert and Casella, theorem 7.8). It should be noted that if a normal prior is used to form the posterior target density, then the conditions for uniform ergodicity are met unless the likelihood is unbounded.

For problems of low dimension with very regular likelihoods, the independence Metropolis chain can be very useful. However, for higher dimensional problems, the same problems which plague importance sampling apply to the independence Metropolis chain. Even if the dominance condition holds, the proposal distribution can miss where the target distribution has substantial mass and can give rise to misleading results. For this reason, independence Metropolis chains are primarily useful in Hybrid samplers (see section 3.10), in which the independence chain is imbedded inside of a Gibbs Sampler.

3.9.2 Random Walk Metropolis Chains

A particularly appealing Metropolis algorithm can be devised by using a random walk to generate proposal values.

(3.10.4) $$\vartheta = \theta + \varepsilon$$

This proposal corresponds to the proposal transition function, $q(\theta,\vartheta) = q_{\varepsilon}(\vartheta - \theta)$ which depends only on the choice of the density of the increments. This proposal function is symmetric, $q(\theta,\varphi) = q(\varphi,\theta)$. A "natural" choice might be to make the increments normally distributed.[24]

GAUSSIAN RANDOM WALK METROPOLIS

```
    Start with $\theta_0$
    Draw $\vartheta = \theta + \varepsilon \quad \varepsilon \sim N(0, s^2\Sigma)$
    Compute $\alpha = \min\left\{1, \frac{\pi(\vartheta)}{\pi(\theta)}\right\}$
    With prob $\alpha$, $\theta_1 = \vartheta$, ELSE $\theta_1 = \theta_0$
    Repeat, as necessary
```

The simplicity of this algorithm gives it great appeal. It can be implemented for virtually any model and it does not appear to require the same in-depth a priori knowledge of π as the independence Metropolis does. In addition, there is an intuitive argument that the random walk algorithm should work better than the independence Metropolis in the sense that it might roam more freely in the parameter space (due to the drifting behavior of a random walk) and "automatically" seek out areas where π has high mass and then navigate those areas. From a theoretical point of view, however, strong properties such as uniform ergodicity cannot be proven for the RW Metropolis without further restrictions on the tail behavior of π. Jarner and Tweedie (2001) show that exponential tails of π is a necessary condition for geometric ergodicity[25] irrespective of the tail behavior of the proposal density q. While there is research into the relative tail behavior of the target and proposal density, there are few definitive answers. For example, it is debatable whether it is desirable to have the proposal (increment) density have thicker tails than the target. There are some results suggestive that this can be helpful in obtaining faster convergence but only for special cases (see Robert and Casella, theorem 6.3.6 which applies only to log-concave targets and Jarner and Tweedie which only applies to univariate problems). In almost all practical applications, a Gaussian Random Walk is used. Again, these theoretical discussions are not terribly relevant to practice since we can always use a normal prior which will provide exponential behavior for the target density.

The problem with the Random Walk Metropolis is that it must be tuned by choosing the increment covariance matrix. This is often simplified by choosing Σ to be either I or the asymptotic covariance matrix. Computation of the asymptotic covariance matrix can be

[24] Interestingly enough, the original paper by Hastings includes examples in which the increments are uniformly distributed.

problematic for models with non-regular likelihoods. We must also choose the scaling factor, s. The scaling factor should be chosen to insure that the chain navigates the area where π has high mass while also producing a chain that is as informative as possible regarding the posterior summaries of interest. These goals cannot be met without some prior knowledge of π which removes some of the superficial appeal of the random walk Metropolis. Even so, the random walk Metropolis has found widespread application and appears to work well for many problems.

3.9.3 Scaling of the Random Walk Metropolis

The scaling of the Random Walk Metropolis is critical to its successful use. If we scale back on the variance of the increments by taking small values of the scaling factor, s, then we will almost always "move" or accept the proposed draw. This will mean that the algorithm will produce a chain that behaves (locally, at least) like a random walk. The chain will exhibit extremely high autocorrelation and, therefore, can provide extremely noisy estimates of the relevant posterior quantities. Given that small values of the scale factor will result in a slowly moving chain which can fail to properly explore regions of high target mass, we might take the opposite approach and take scale factors considerably larger than one. This can result in a chain proposed moves which are far away from the region of posterior mass (overshoot the target) and a high rejection rate. In the extreme case, over-scaled RW Metropolis chains can get stuck.

"Optimal" scaling of the Random Walk Metropolis is a balancing act in which the scaling factor chosen to optimize some performance criterion. A reasonable criterion is to measure information obtained about a posterior quantity per unit of computing time. Since we are using averages of a function of the draws to approximate the integral of that function

[25] Roughly speaking, geometric ergodicity means that the distance between the n-step kernel and the invariant

with respect to the target posterior, the reciprocal of the variance of the sample average can be used as a measure of information. This clearly depends on the particular function used. Given a choice of function (such as the identity function for posterior means), we must be able to calculate the variance of the sample average of this function where the sample comes from a stationary but autocorrelated process. Let $\mu = E_\pi\left[g(\theta)\right]$, then we estimate μ with $\hat{\mu} = \frac{1}{R}\sum_r g\left(\theta^r\right) = \frac{1}{R}\sum_r g^r$.

(3.10.5)
$$\operatorname{var}(\hat{\mu}) = \frac{1}{R^2}\begin{bmatrix} \operatorname{var}\left(g^1\right) + \operatorname{cov}\left(g^1, g^2\right) + \ldots + \operatorname{cov}\left(g^1, g^R\right) + \\ \operatorname{cov}\left(g^2, g^1\right) + \operatorname{var}\left(g^2\right) + \ldots + \operatorname{var}\left(g^R\right) \end{bmatrix}$$

Since $\left\{g^r\right\}$ is a stationary process, we write (3.10.5) using the autocorrelations of the process.

(3.10.6)
$$\operatorname{var}(\hat{\mu}) = \frac{\operatorname{var}(g)}{R}\left[1 + 2\sum_{j=1}^{R-1}\left(\frac{R-j}{R}\right)\rho_j\right] = \frac{\operatorname{var}(g)}{R} f_R$$

The "price" of autocorrelation in the MCMC draws is represented by the factor f_R which is the multiple by which the variance is increased over the estimate based on an iid sample. Some use the reciprocal of f as a measure of the *Relative Numerical Efficiency* of the sampler.

In practice, we can use the sample moments of the g process to estimate (3.10.6). Some guidance is required for the choice of the number of autocorrelations to include in the computation of f. This is subject of a considerable literature in time series. We shouldn't take this formula literally and use all R-1 computable autocorrelations. This would create a noisy and inconsistent estimate of f. There are two strategies in the time series literature. Some put a "taper" or declining weights on the sample autocorrelations used in estimating f. Others simply advocate truncating the sum at some point m.

distribution decreases to the power of n.

(3.10.7) $$\hat{f}_R = 1 + \sum_{j=1}^{m} \left(\frac{m+1-j}{m+1} \right) \hat{\rho}_j$$

Some guidance is required in the choice of m. In order for the estimator to be consistent, we must promise to increase m as the sample size (R) increases (albeit at a slower rate than R). This still does not provide guidance in the choice of m for a fixed sample size. Many MCMC applications will typically involve 10,000 or more draws. The autocorrelation structure of many chains used in practice is complicated and can have significant autocorrelation out to lags of order 100 or more. The exponential decline in autocorrelation associated with linear time series models is often not present in the MCMC chains. For these reasons, m should be chosen to be at least 100.

Given a measure of the information content, we can consider the problem of optimal scaling of the RW Metropolis chain. That is, we can optimize the choice of s to maximize relative numerical efficiency or to minimize f. Clearly, the optimal choice of s depends on the target distribution. Gelman et al (1996) and Roberts and Rosenthal (2001) provide some results on optimal scaling but only for target distributions that are products of identical normal densities. They also consider the interesting asymptotic experiment of increasing the dimension of the parameter vector. This analysis presents both the optimal scaling factor as well as the acceptance rate of the optimally scaled RW chain. For the case of a target distribution consisting of only one normal univariate density, the optimal scaling factor is s= 2.38. As the dimension increases, an asymptotic result has the scaling reduced at the rate of the square root of the dimension, $s = 2.3/\sqrt{d}$. Corresponding to these scaling results are implied optimal acceptance rates which are around .23.[26] In practice, it is

[26] Gelman et al (96) refer to optimal acceptance rates of closer to .5 for d=1. This is because this paper only considered the first order autocorrelation. This has led to the incorrect interpretation that the optimal acceptance rate declines from around .5 to .23 or so as d increases.

impossible to determine how differences between the target for our problem and an iid normal distribution will translate into differences in optimal scaling. It is true that we are scaling the asymptotic covariance matrix so that if the target density is close to a normal with this covariance matrix then we can expect the Roberts and Rosenthal results to apply. However, if our target density differs markedly from the asymptotic normal approximation, it is possible that the optimal scaling is quite different from this rule of thumb. We recommend that, where possible, shorter runs of the RW chain be used to help tune or choose the scaling constant to maximize numerical relative efficiency for parameters of interest. Of course, this assumes that the researcher has already determined the "burn-in" period necessary to dissipate initial conditions. We will provide some guidelines on selection of burn-in period in section 3.11. Contrary to current practice, choice of scaling should not be made on the basis of the acceptance rate of the chain but rather on the measure of numerical efficiency which is the more directly relevant quantity.

3.10 Metropolis Algorithms Illustrated with the Multinomial Logit Model

As discussed in chapter 2, the Multinomial Logit Model is arguably the most frequently applied model in marketing applications. Individual data on product purchase often has the property that individuals are seldom observed to purchase more than one product of a specific type on one purchase occasion. The logit model has also been applied in a variety of forms to aggregate market share data. The MNL model has a very regular log-concave likelihood but it is not in a form that is easily summarized. Moments of functions of the parameter vector are not computable using analytic methods. In addition, the natural conjugate prior is not easily interpretable so that it is desirable to have methods which would work with standard priors such as the normal prior. If we assess a standard normal prior, we can write the posterior as

$$\begin{aligned} &\pi\left(\beta \middle| X, y\right) \propto \ell\left(\beta \middle| X, y\right)\pi\left(\beta\right) \\ (3.11.1) \qquad &\pi\left(\beta\right) \propto \left|A\right|^{1/2} \exp\left\{-\tfrac{1}{2}\left(\beta - \bar{\beta}\right)' A\left(\beta - \bar{\beta}\right)\right\} \end{aligned}$$

The likelihood in (3.11.1) is just the product of the probabilities of the observed choices or discrete outcomes over the n observations.

$$\begin{aligned} &\ell\left(\beta \middle| X, y\right) = \prod_{i=1}^{n} \Pr\left(y_i = j \middle| X_i, \beta\right) \\ (3.11.2) \qquad &\Pr\left(y_i = j \middle| X_i, \beta\right) = \frac{\exp\left(x_{i,j}'\beta\right)}{\sum_{j=1}^{J} \exp\left(x_{i,j}'\beta\right)} \end{aligned}$$

y is a vector with the choices (1, … J) and X is an nJ x k matrix of the values of the x variables for each alternative on each observation.

Experience with the MNL likelihood is that the asymptotic normal approximation is excellent. This suggests that Metropolis algorithms based on the asymptotic approximation will perform extremely well. We also note that the MNL likelihood has exponential tails (even without the normal prior) and this should provide very favorable theoretical convergence properties. We implement both an independence and RW Metropolis algorithm for the MNL model. Both Metropolis variants use the asymptotic normal approximation.

$$(3.11.3) \qquad \pi\left(\beta \middle| X, y\right) \dot{\propto} \left|H\right|^{1/2} \exp\left\{\tfrac{1}{2}\left(\beta - \hat{\beta}\right)' H\left(\beta - \hat{\beta}\right)\right\}$$

We have a number of choices for $\hat{\beta}$, H. We can simply use the MLE for $\hat{\beta}$ or we could find the posterior mode (preferable for truly informative priors) at about the same computational cost. H could be minus the actual Hessian of the likelihood (alternatively, the posterior) evaluated at $\hat{\beta}$ or the sample information matrix. Alternatively, we can use expected sample information which can be computed for the MNL.

$$
\begin{aligned}
&H = -E\left[\frac{\partial^2 \log \ell}{\partial\beta\partial\beta'}\right] = \sum_i X_i A_i X_i^{'} \\
(3.11.4) \qquad & X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}; \quad A_i = \text{Diag}(p_i) - p_i p_i^{'}
\end{aligned}
$$

p_i is a J vector of the probabilities for each alternative for observation i.

We illustrate the functioning of Metropolis algorithms using a small sample of simulated data, chosen to put the asymptotic approximations to a severe test. N=100, J=3, and k=4 (two intercepts and two independent variables that are produced by iid unif(0,1) draws). We set the beta vector to (-2.5, 1.0, .7, -.7). The first two elements of β are intercepts for alternatives 2 and 3 that are expressed (in the usual manner) relative to alternative 1 which is set to 0. These parameter settings imply that the probability of alternative 2 will be very small. In our simulated sample, we observed alternative 1 26 times, alternative 2 only 2 times and alternative 3 72 times.

To implement the independence Metropolis, we use a MVst candidate sampling distribution. That is, we draw candidate parameter vectors using $\beta \sim \text{MVst}\left(\upsilon, \hat{\beta}, H^{-1}\right)$. The only "tuning" required is the choice of ν. Too small values of ν will be inefficient in the sense of producing such fat tails that we will reject draws more often than for smaller values of ν. In addition, very small values of ν such as 4 or less produce a distribution which has fat tails but is also very "peaked" without the "shoulders" of the normal distribution. This also implies that the Metropolis algorithm would suffer inefficiencies from repeating draws to build-up mass on the shoulders of the peaked t distribution. Clearly, one could tune the independence Metropolis by picking ν so as to maximize a numerical efficiency estimate or minimize f in (3.10.7). Using the mean of the parameters as the function whose numerical efficiency should be assessed, we find that numerical efficiency is relatively flat in the range of 5 to 15. Very small values of ν result in only slightly reduced numerical efficiency. All results reported here are for $\nu=6$. `rmnlIndepMetrop` provides the R implementation of this algorithm (available in the package, ***bayesm***).

The Random Walk Metropolis must be scaled in order to function efficiently. In particular, we propose β values using the equation

$$\beta_{\text{cand}} = \beta_{\text{old}} + v \qquad v \sim N\left(0, s^2 H^{-1}\right) \tag{3.11.5}$$

According to the Roberts and Rosenthal guidelines, s values close to $2.93\big/\sqrt{d} = 2.93\big/2$ should work well. Given the accuracy of the normal approximation, we might expect the Roberts and Rosenthal guidelines to work very well since we can always transform the asymptotic normal approximation into a product of identical normal densities. However, the result is "asymptotic" in d. Figure 3.14 shows numerical efficiencies as measured by the square root of f as a function of the scaling constant s. We use the square root of f as we are interested in minimizing the numerical standard error and the square root of f is the multiple of the iid standard error. Each curve in figure 3.14 shows numerical efficiency for each of the four parameters and "tuning" runs of 20,000 iterations. We consider only the estimation of the posterior mean in figure 3.14. The small inset figure shows the acceptance rate as a function of s. As s increases, we expect the acceptance rate to decline as the chain navigates more freely. However, the numerical efficiency is ultimately the more relevant criterion. Numerical efficiency is maximized at around 1.25 which is close to the Roberts and Rosenthal value of 1.47. It is worth noting that minimum numerical efficiency of the RW Metropolis is around 4.

In comparison to the RW, the independence Metropolis functions much more efficiency with an acceptance rate of .70 and a numerical efficiency of 1.44 or barely less than iid sampling. Figure 3.15 shows the estimated posterior distribution of β_1 constructed from 50,000 draws of the independence and RW Metropolis chains. To the right of the distribution are the corresponding ACFs. Differences between the ACFs of the RW and Independence chains result in a numerical efficiency ratio of 3:1. The light green line on the distributions is the diffuse normal prior. The solid density curve is the asymptotic normal approximation to the posterior. Even in this extreme case, the actual posterior is only slightly skewed to the left from the asymptotic approximation.

For highly correlated chain output, there is a practice of "thinning" the output by selecting only every mth draw. The hope is that the "thinned" output will be approximately iid so that standard formulas can be used to compute numerical standard errors. Not only is this practice unnecessary in the sense that the investigator is literally throwing away information, but it can be misleading. In the case of the RW chain here, keeping every 10th observation (a very standard practice) will still produce a chain with non-trivial autocorrelation. The only possible reason to thin output is for convenience in storage. Given that correct numerical standard errors are trivial to compute, it seems odd that this practice continues.

3.11 Hybrid MCMC Methods

In practice, many problems are solved with a "Gibbs-style" strategy for constructing the Markov Chain. In particular, hierarchical models have a structure that allows for an efficient strategy which involves various conditional draws. This is because of conditional independence assumptions which are employed in constructing hierarchical models. However, in many cases, direct draws from the requisite conditional distribution are not available and some combination of a Gibbs style chain and Metropolis steps must be used. For example, consider the hierarchical structure given below.

(3.12.1) $$\theta_2 \rightarrow \theta_1 \rightarrow \text{Data}$$

The Gibbs sampler for (3.12.1) would consist of

(3.12.2) $$\theta_2 | \theta_1$$

and

(3.12.3) $$\theta_1 | \theta_2, \text{Data}$$

If conjugate priors are used, the draw defined by (3.12.2) is often a direct 1 for 1 draw from the conditional distribution. However, for many models the draw in (3.12.3) can be difficult to achieve. One possible strategy would be to break the draw each of the elements of the θ_1 vector one by one using Griddy Gibbs, sliced sampling or Adaptive Rejection Metropolis Sampling (ARMS, see Gilks et al (1995)). However, in many real problems, the dimension of θ_1 will be large so that this is not practical.

A useful idea is to replace the “Gibbs” draw in (3.12.3) with a Metropolis step. This algorithm is called by some a “Hybrid” chain or “Metropolis within Gibbs.” To construct the Metropolis chain all this is required is to evaluate the conditional density in (3.12.3) up to a normalizing constant. Of course, the joint density evaluated at the draw of θ_2 is proportional to the conditional density and can be easily evaluated.

$$p(\theta_1 | \theta_2, \text{Data}) \propto p(\text{Data} | \theta_1, \theta_2) p(\theta_1, \theta_2)$$

If we implement a Metropolis step with the conditional distribution, $\theta_1 | \theta_2, \text{Data}$, as its invariant distribution, then it can be shown that the hybrid chain has the full posterior as its equilibrium distribution. To see this, we first return to a discussion of the Gibbs Sampler and develop the notion of the composition of two Markov chains.

The standard "two-stage" Gibbs sampler can be written:

(3.12.4) $$\theta_1 | \theta_2, \text{Data}$$

and

(3.12.5) $$\theta_2 | \theta_1, \text{Data}$$

The Markov Chain that (3.12.4) and (3.12.5) represent can be thought of as the combination of two chains, each one of which updates one of the components of θ. If we denote K_1^G as the kernel of a chain formed by the conditional distribution in (3.12.4) and K_2^G as the kernel of the chain formed by (3.12.5), then the two-stage Gibbs Sampler is the composition of these two chains:

(3.12.6) $$K^G = K_1^G \circ K_2^G$$

The Kernel of the composed chain is

(3.12.7) $$\begin{aligned} &K(\theta, A) = \int_A p(\theta, \varphi) d\varphi \\ &p(\theta, \varphi) = p((\theta_1, \theta_2), (\varphi_1, \varphi_2)) = \pi_{1|2}(\varphi_1 | \theta_2) \pi_{2|1}(\varphi_2 | \varphi_1) \end{aligned}$$

While each of the sub-chains are reducible (i.e. they only navigate on the sub-space formed by holding one component of θ fixed), the full chain is irreducible as we have seen before. What is less obvious is that the full posterior is the invariant distribution of both K_1^G and K_2^G, viewed separately. To see this, consider K_1^G. If we start with a draw from the joint posterior, we must then show that the one iteration of K_1^G reproduces the joint posterior. The joint posterior factors into the conditional and marginal densities.

$$\pi(\theta_1, \theta_2 | \text{Data}) = \pi_2(\theta_2 | \text{Data}) \pi_{1|2}(\theta_1 | \theta_2, \text{Data})$$

This means that the "initial" value is a draw that can always be viewed as obtained by first drawing from the appropriate marginal and then drawing from the conditional distribution.

$$\theta_2^0 \sim \pi_2 \text{ and } \theta_1^0 \sim \pi_{1|2}$$

To show that K_1^G produces a draw from the joint posterior, we observe that

$$\theta_1^1 \sim \pi_{1|2}\left(\theta_2^0\right)$$

The notation, $\pi_{1|2}\left(\theta_2^0\right)$, means the conditional distribution of $\theta_1 \middle| \theta_2 = \theta_2^0$. Thus, the draw from K_1^G is also a draw from the joint.

$$\left(\theta_1^1, \theta_2^0\right) \sim \pi_{1|2}\pi_2 = \pi_{1,2}$$

The same argument applies to K_2^G and we can see that the full two-stage Gibbs Sampler is irreducible and has the joint posterior as its variant distribution.

The hybrid chain formed by substituting a Metropolis step for first step in the two-stage Gibbs sampler is also the composition of two reducible chains, each having the joint posterior as its invariant distribution. $K^H = K_1^M \circ K_2^G$. This follows directly by exactly the same argument that we applied to the Gibbs sampler. There is one additional important point. The Metropolis step will update the value of θ_1 given the previous iteration value of θ_2. This means that the values of θ_2 from the previous iteration can be used in the computation of candidate or proposal values of θ_1 in the Metropolis step. As long as we obey the proper conditioning and use only values of the *last* iteration, we will still have a Markov chain with the correct invariant distribution. This means that we can "automatically" adjust the Metropolis candidate sampling density depending on the last value of θ_2. We should also note that this is not the same as "adaptive" schemes that use information from a past subsequence to adjust the Metropolis step. These schemes are not Markovian and require additional analysis to establish that they have the proper invariant distribution.

3.12 Diagnostics

The theoretical properties of MCMC methods are quite appealing. For many algorithms and problems, especially with normal priors , it is easy to establish ergodicity and even stronger results such as geometric rates of convergence and uniform ergodicity. The problem with these rates of convergence results is that they do not specify the constants that govern the actual rate of convergence. In practice, we can produce an MCMC sampler which has desirable theoretical properties but poor performance given our finite computer resources. In addition, errors in formulating the model or the MCMC algorithm can be very important and difficult to diagnose.

The performance of MCMC algorithms is related to the speed at which the chain navigates the state space. Highly autocorrelated samplers require long periods of time to navigate the parameter space fully enough to properly represent the level of uncertainty in the posterior. A related problem is the dissipation of initial conditions. In practice, we start our samplers from reasonably arbitrary points and hope that they dissipate the effect of these initials after a burn-in period. Draws after the burn-in period are viewed as draws from the stationary distribution. Clearly the speed of navigation and dissipation of initial conditions are related. Near iid MCMC samplers will both dissipate initial conditions and rapidly navigate the parameter space. However, in practice we see examples of samplers that dissipate the initial conditions rapidly but are slow to navigate regions of high posterior mass. For example, if we consider the bivariate normal Gibbs sampler, high (much higher than .95) correlation will create a narrow ridge which the Gibbs sampler may require many iterations to navigate. However, given the thin tails of the normal distribution, if we start the sampler at a point far from this ridge, the sampler will quickly move to the ridge and dissipate the initial condition. The classic "witch's hat" examples are examples where a Gibbs sampler can get stuck in one mode of a very peaked distribution and fail to break out to the other mode even though this is theoretically possible. In our experience with models relevant to micro data and marketing applications, these situations are rarely encountered. More common are situations in which the MCMC sampler navigates slowly with a high degree of autocorrelation. In these situations, it may take days of computing to properly navigate the posterior.

Unfortunately, there is a sense in which there can be no powerful diagnostic or "test" for failure of convergence without substantial prior knowledge of the posterior. If the asymptotic approximation to the posterior is poor and/or if the parameter space is of high dimensions, then we are engaging in MCMC sampling in order to learn about the posterior. This means that we can never be very sure that we have gotten the right result. Many of the proposals for convergence diagnostics utilize a sub-sequence of an MCMC run in order to assess convergence. This requires confidence that the sampler has already navigated the relevant regions of the parameter space and the issue is more one of the information content of the sampler sequence and not whether or not navigation is complete. The information content of the sequence can easily be gauged by computing numerical standard errors for quantities of interest using the formula in (3.10.7). This doesn't address the convergence question. Other proposals in the convergence diagnostics areas involve starting the sampler from a number of initial conditions (Gelman and Rubin (1992) to check for dissipation of initial conditions. Again, choice of these initial conditions requires information about the posterior. In problems where the parameter space is more than a few dimensions, it may be impractical to choose a comprehensive set of initial conditions. Others have suggested using multiple "parallel" runs of the MCMC sampler to compare the distribution obtained within one run with the cross-section simulation of many runs. This is not practical in the large scale problems considered here.

In practice, we rely on sequence or time series plots of MCMC output as well as computed autocorrelation and associated standard errors to monitor convergence. The assumption is that a slowly navigating chain (one with near random walk behavior) must be monitored very closely for possible convergence failure. These highly autocorrelated chains should be run as long as possible to investigate the sensitivity of the estimation of the posterior variability to run length. In many situations, the MCMC sampler may exhibit dependence of an extremely long lasting or persistence variety, often with autocorrelations that do not damp off quickly. In these situations, more attention should be focused on monitoring convergence. We have found that one of the most effective ways to assess convergence properties is to conduct sampling experiments that have been calibrated to match the characteristics of the actual data under analysis. While large scale sampling experiments may not be possible due to the size of the parameter space or due to computing limitations, we have found that a small scale experiment can be very useful.

In most situations, slowly navigating chains are produced by high correlation between groups of parameters. The classic example of this is the normal Gibbs sampler. Other examples include latent variables added for data augmentation. Here high correlation between the latent variables and the model "parameters" can produce slowly navigating chains. There are two possible solutions to high correlation between two subsets of parameters. If the two subsets can be "blocked" or combined into one iid draw, then the autocorrelation can be reduced substantially. However, care must be exercised in blocking as there are examples where the blocked sampler gets stuck. If one set of parameters is not of direct interest and can be integrated out (sometimes called "collapsing"), then the autocorrelation can also be reduced (see Liu (2001) for a theoretical comparison of collapsed and blocked samplers). We will see an example of this technique in the case study on scale usage heterogeneity. Of course, whether these are practical strategies depends on the model and priors and our ability to perform the integration. Finally, we have seen that, with a good proposal density, the Independence Metropolis can outperform the Random Walk.

One neglected, but important, source of concern are errors made in the formulation and coding of an MCMC method. One can easily formulate an MCMC sampler with the wrong posterior for the model under consideration. In addition, errors can be made in the coding of the densities used in Metropolis methods or in the draws from the conditional distributions. While simulation experiments will often detect the presence of these errors, Geweke (2004) has proposed an additional useful check on the formulation of MCMC samplers. The idea is to simulate from the joint distribution of the observables and the parameters in two independent ways. One way will use the MCMC method to be tested and the other will simply require drawing from the prior. Draws made in these two ways can then be compared to see if they are similar.

To draw from the joint distribution of the observations and parameters, we simply need the ability to simulate from the prior and the model.

$$p(y,\theta)=p(\theta)p(y|\theta) \tag{3.13.1}$$

IID draws from this joint density can be achieved by drawing from the prior and then drawing $y|\theta$ from a model simulator. Code to simulate from the prior and the model is not usually needed to implement the MCMC method[27]. This is the sense that coding errors in the MCMC method could be independent of coding errors in the direct draws from (3.13.1). To indirectly draw from the joint distribution of the observables and θ, we can use the MCMC method to draw from the posterior in conjunction with the model simulator. These two draws can be used to construct a hybrid chain with the joint distribution of (y,θ) as the invariant distribution.

EXTENDED SAMPLER

Start from some value θ^0

Draw $y^1\left|\theta^0\right.$ using the model simulator

Draw $\theta^1\left|y^1,\theta^0\right.$ using the MCMC method to be tested

Repeat as necessary

Denote the sequence of iid draws as $\left\{y^r,\theta^r\right\}$ and the sequence from the hybrid sampler as $\left\{\hat{y}^r,\hat{\theta}^r\right\}$. Geweke suggests that we compare $\bar{g}=\frac{1}{R}\sum_r g\left(y^r,\theta^r\right)$ with $\bar{\bar{g}}=\frac{1}{R}\sum_r g\left(\hat{y}^r,\hat{\theta}^r\right)$ as a diagnostic check on the MCMC algorithm. Differences between $\bar{g}$ and $\bar{\bar{g}}$ could represent conceptual errors in the MCMC algorithm which result in a different invariant distribution than the posterior of this model and prior or coding errors in implementing various aspects of the MCMC algorithm. Clearly, the power of this diagnostic to detect errors depends on the scope of the g function. There will be a trade-off between power for certain types of errors versus ability to perform omnibus duty against a

[27] For conditionally conjugate models, draws from the prior are usually achieved by using the same functions required to draw from the conditional posteriors (e.g. normal and Wishart draws). Thus, the Geweke method will not have "power" to detect errors in these simulation routines. However, simulating from the model is truly an independent coding task.

wide spectrum of errors. In particular, g does not have to be a function of the observables at all. In many contexts, investigators use very diffuse, but proper priors, these prior hyperparameter settings will result in low power for the Geweke method. In order to obtain better power, it is useful to tighten up the priors by using hyper-parameter settings very different from those used in analysis of data. Finally, we prefer to compare the entire distribution rather than the particular moments implied by the g function (at the risk of focusing on univariate marginals). For example, qqplots using elements of (y,θ) drawn by the two methods can be very useful.

Figure 3.1
Functioning of Bivariate Normal Gibbs Sampler

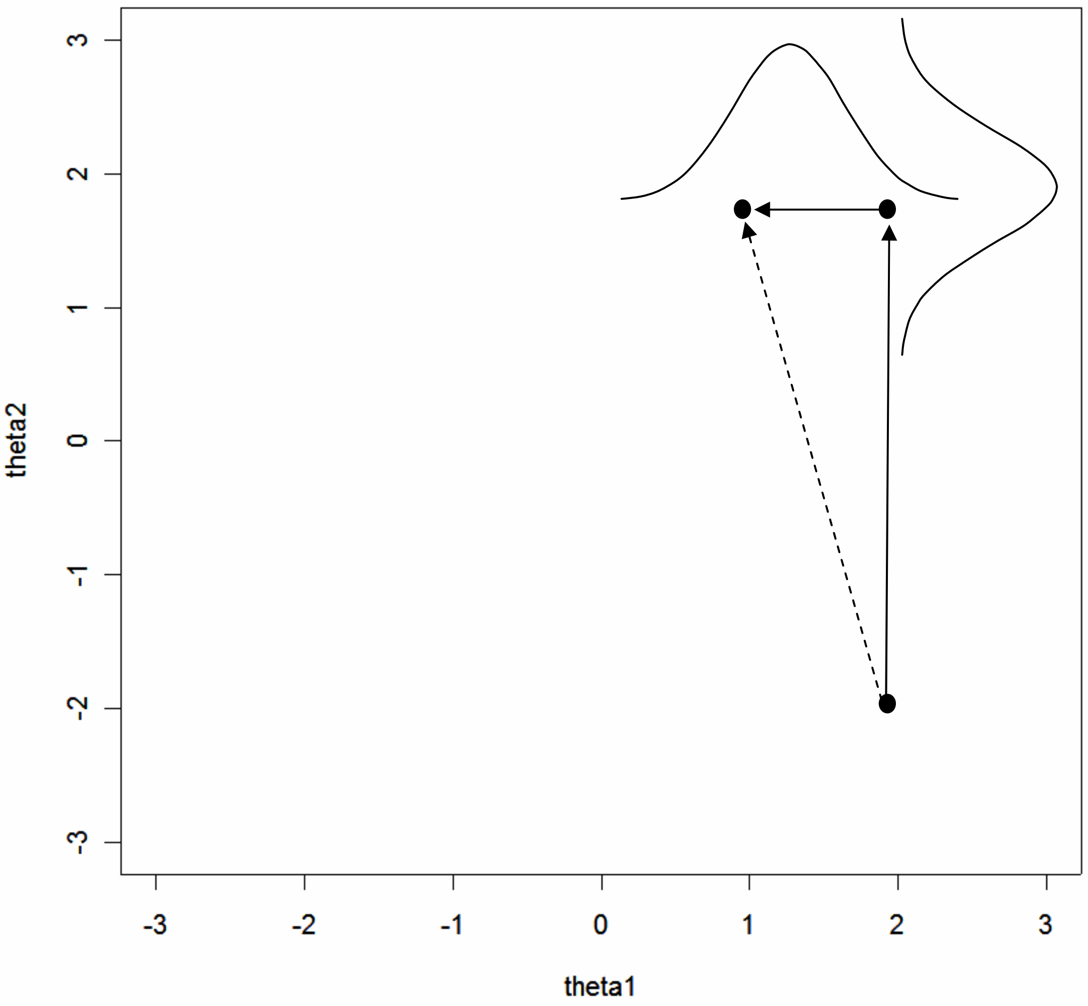


Figure 3.2
20 Draws from Bivariate Gibbs Sampler Showing Intermediate Moves

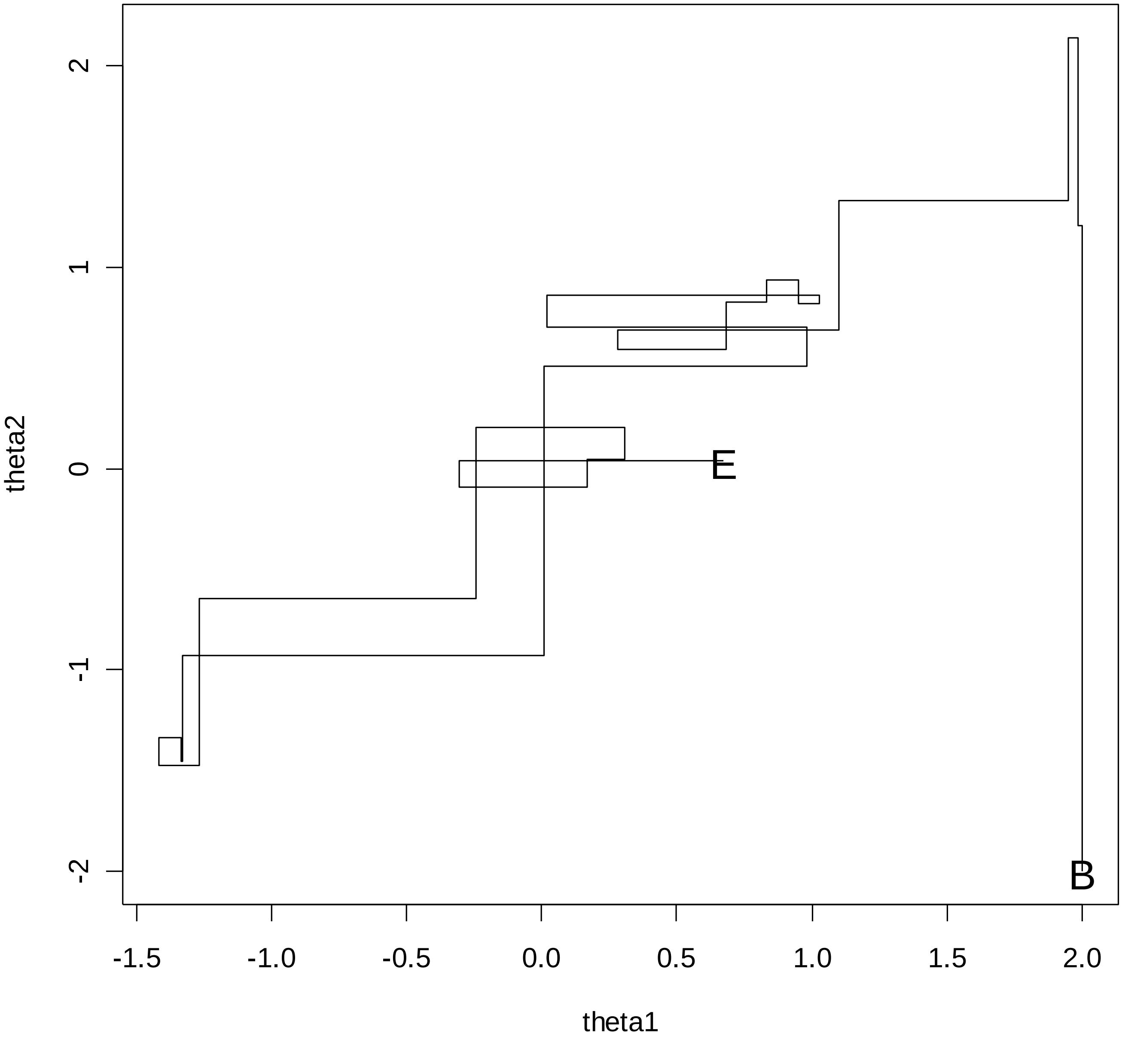


Figure 3.3
1000 Draws from Bivariate Gibbs Sampler

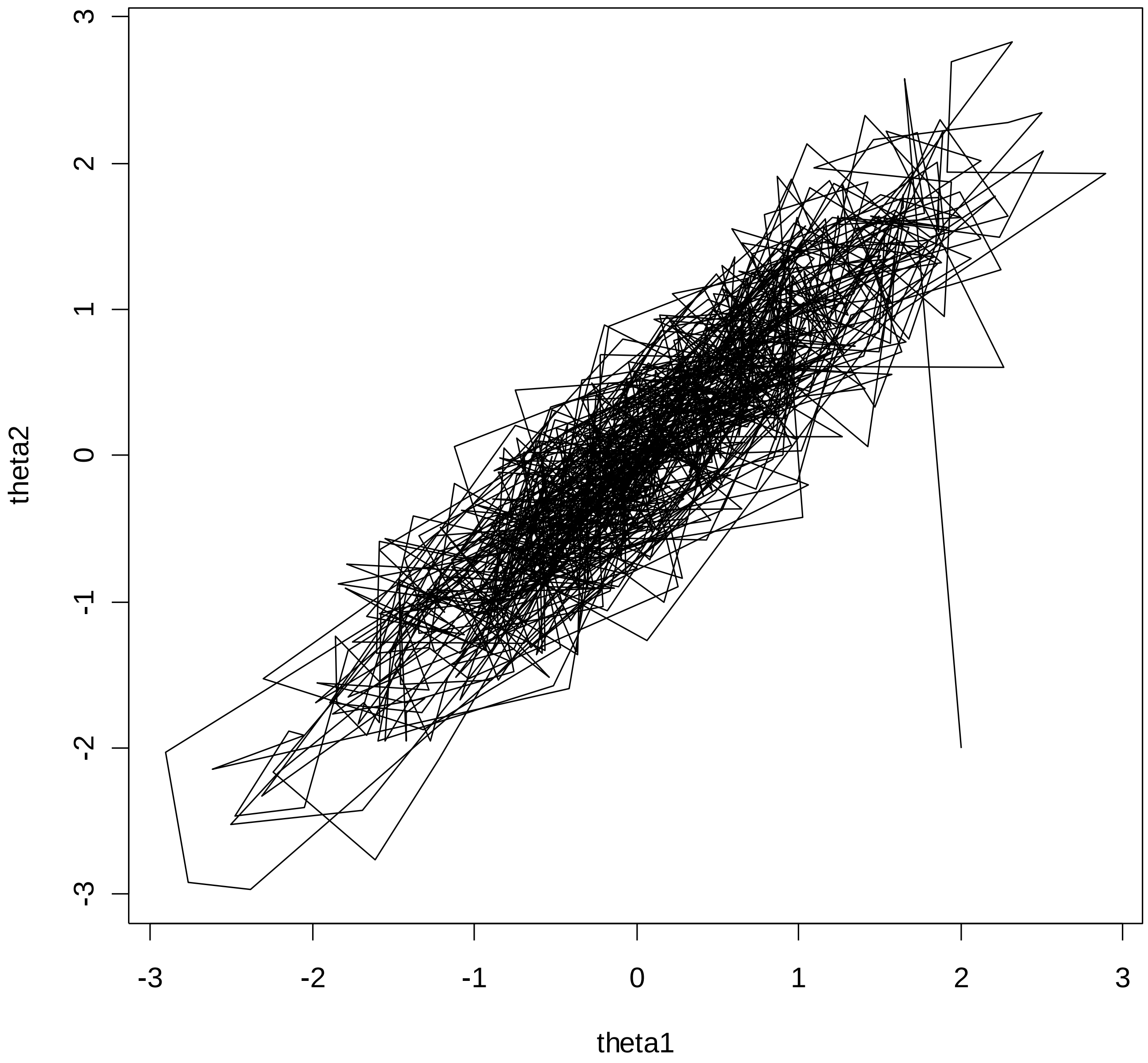
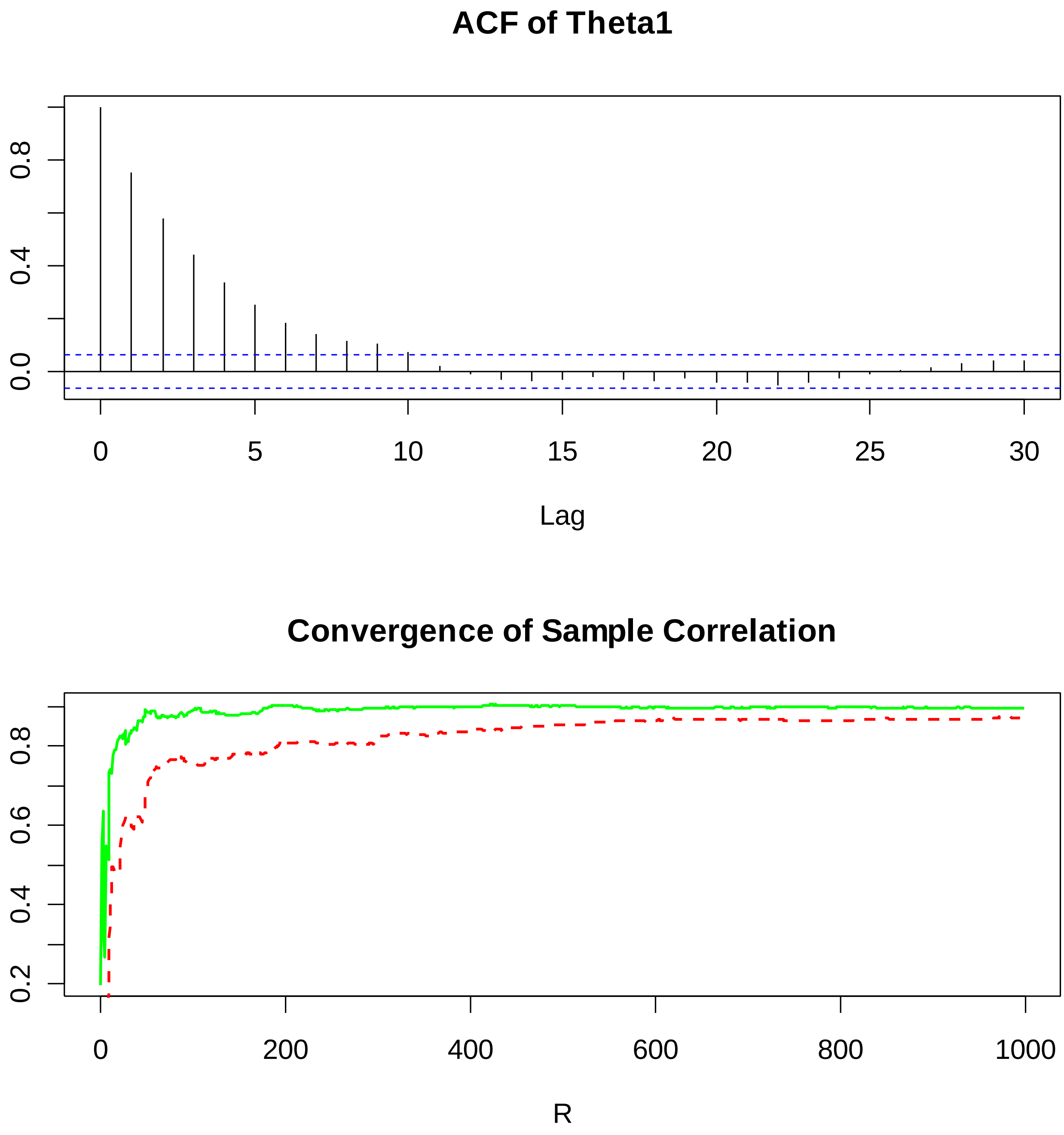


Figure 3.4
Autocorrelation Function and Illustration of Ergodicity



ACF of Theta1

Convergence of Sample Correlation

Figure 3.5
Double Exponential Density and Discrete Approximation

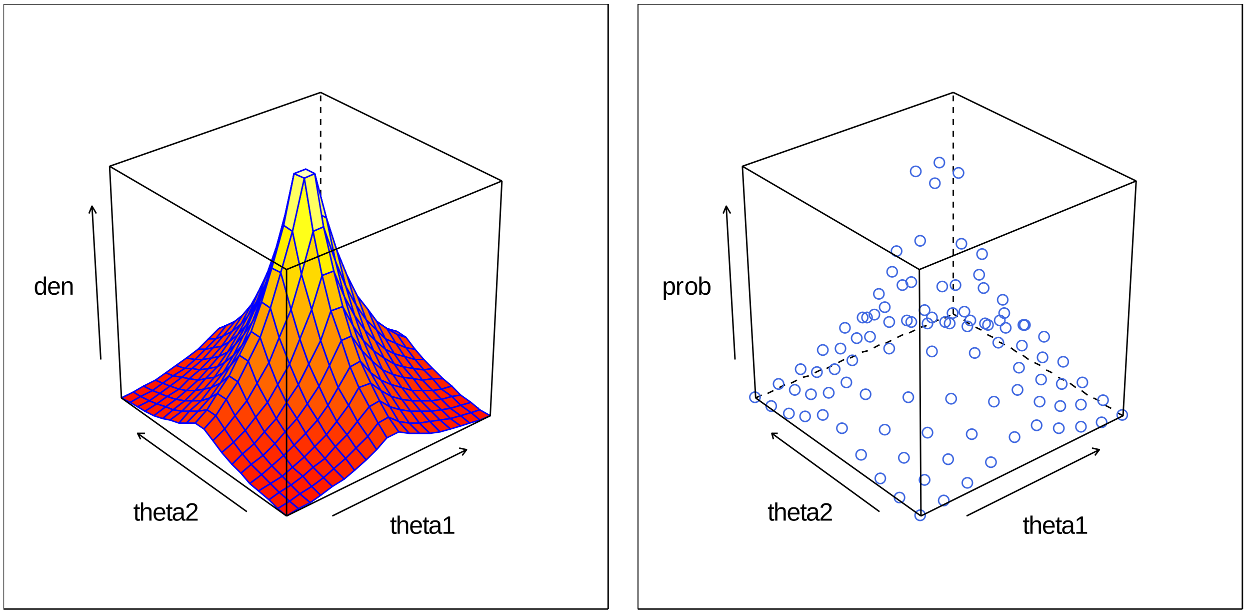


Figure 3.6
Linear Hierarchical Models: Failure of Least Squares Estimates

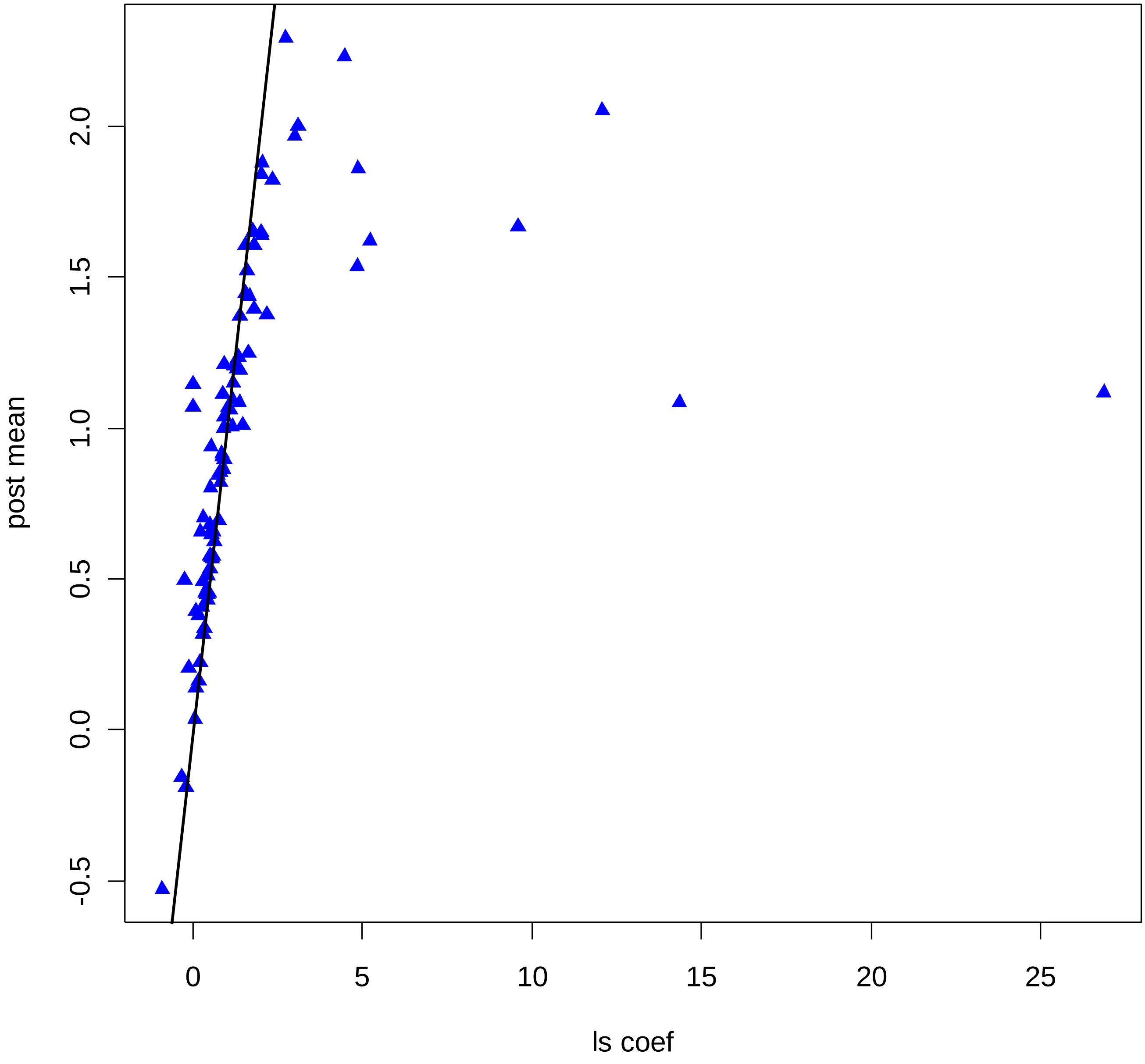


Figure 3.7
Illustration of Shrinkage in Hierarchical Linear Models

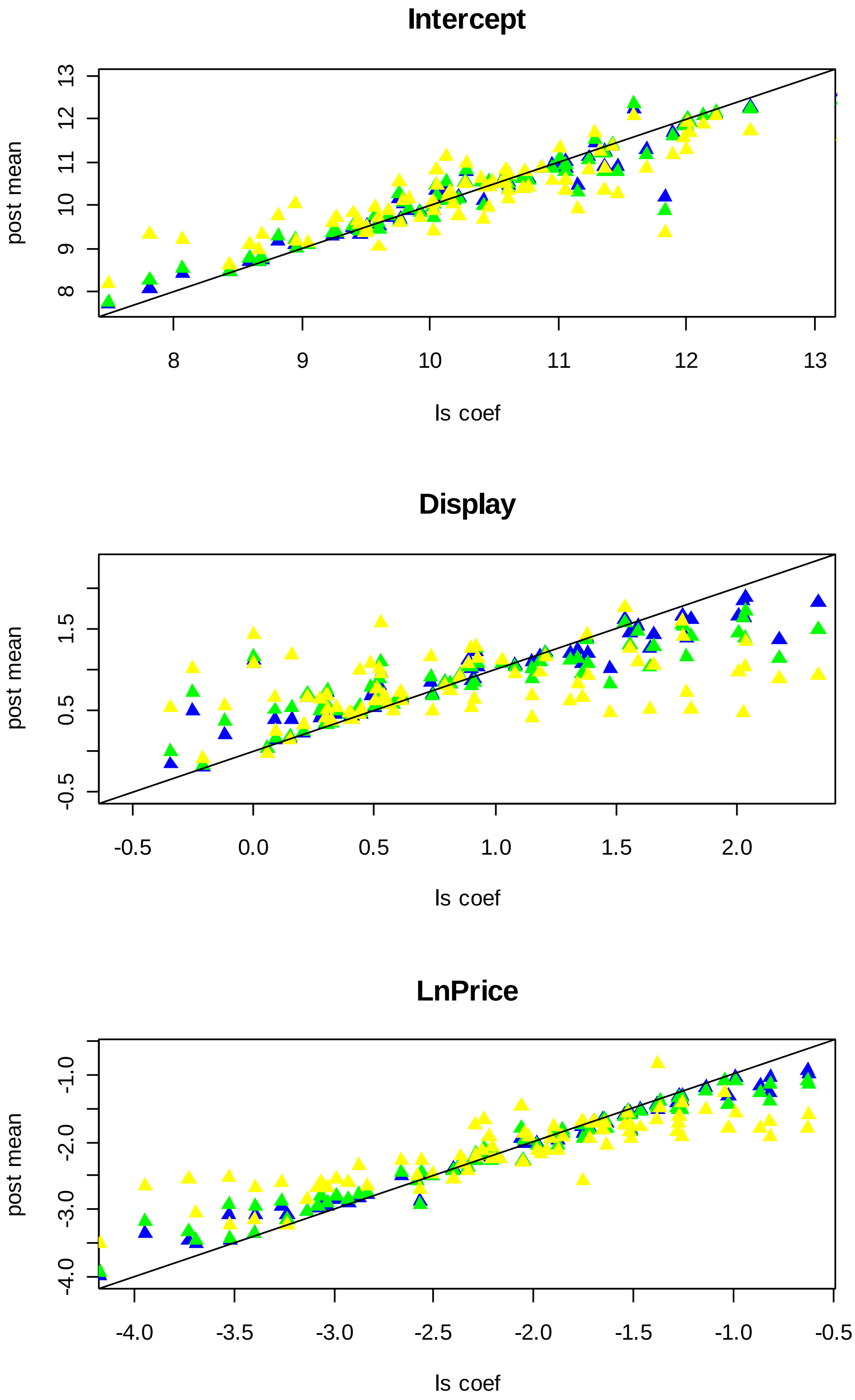


Figure 3.8
Gibbs Sampler Results for Binary Probit

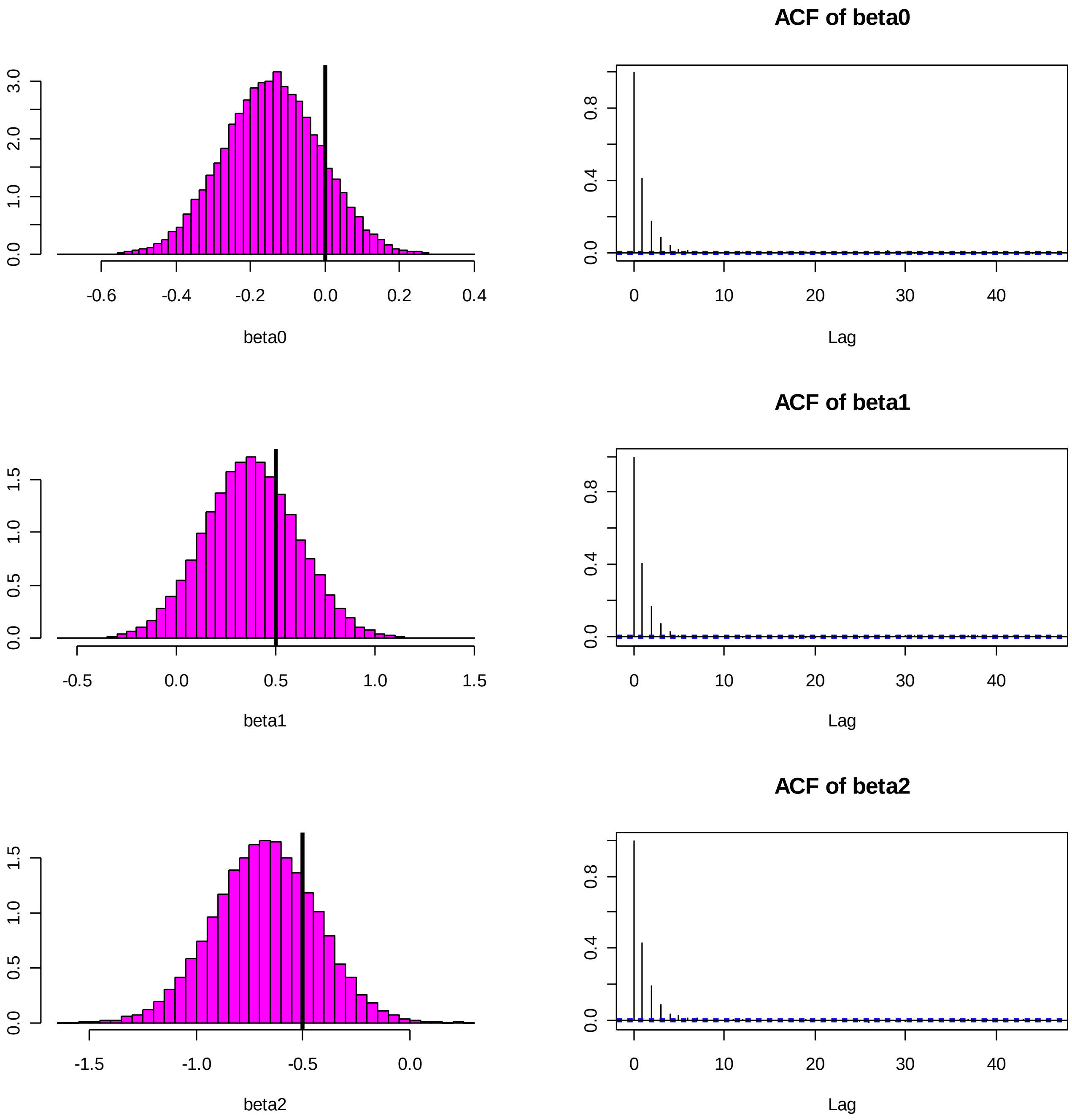


Figure 3.9
Non-normality of Probabilities

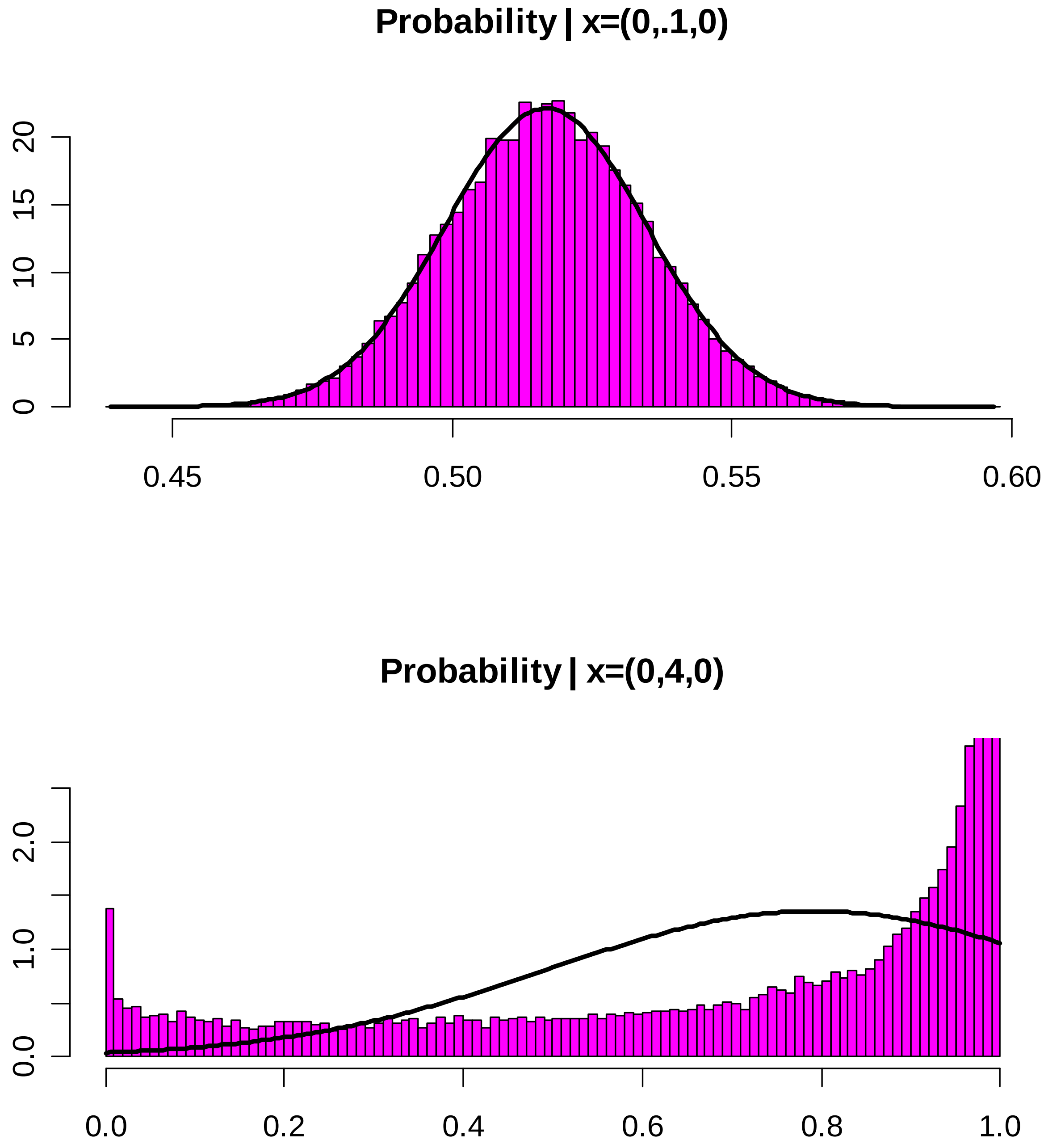


Figure 3.10
Illustration of Label-Switching for Mixture of Normals Model

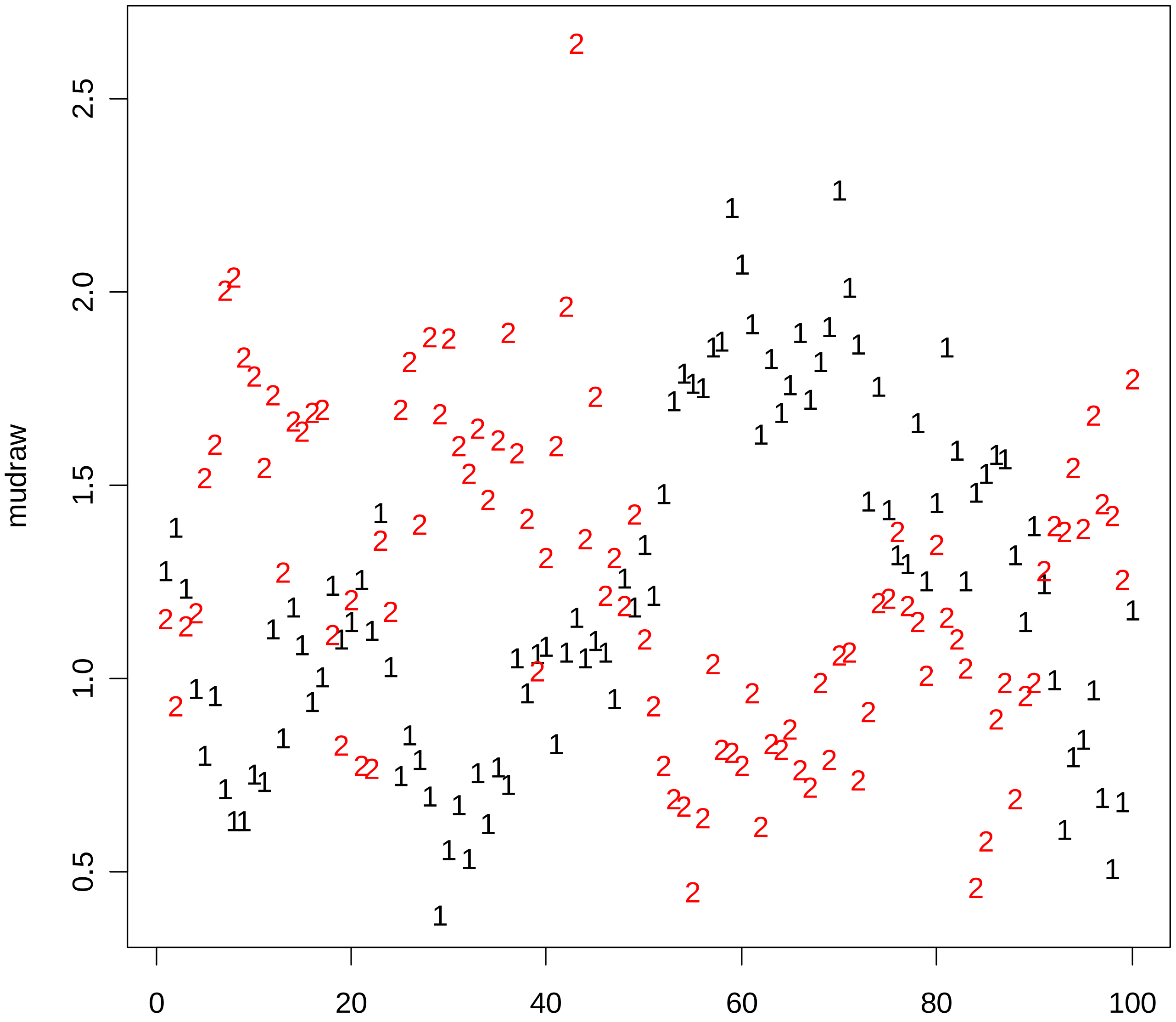


Figure 3.11
Posterior Draws of Density:
Example of Univariate Mixture of Two Normals

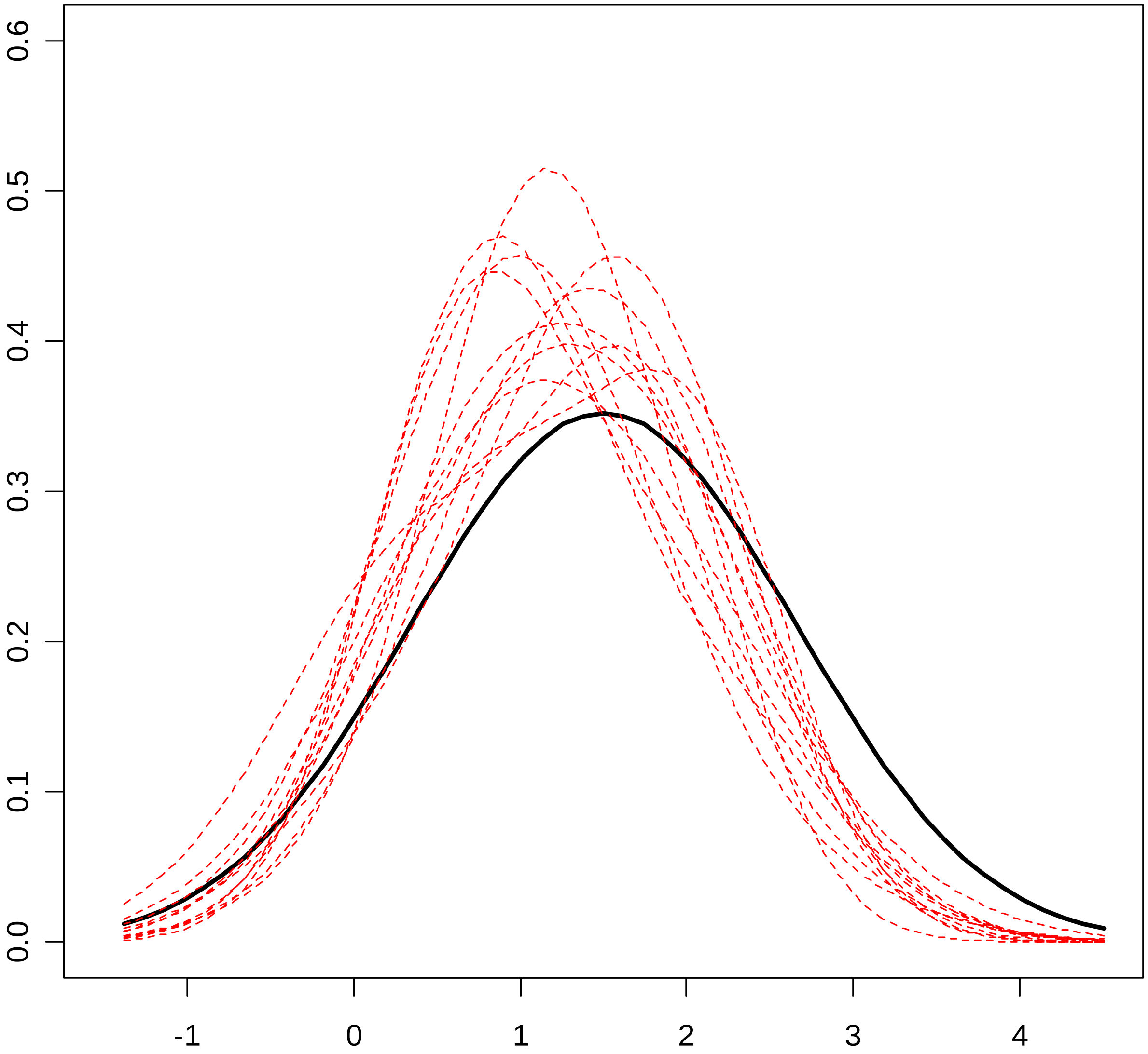


Figure 3.12
Plot of Frequency of Normal Components:
Multivariate Normal Mixture Example

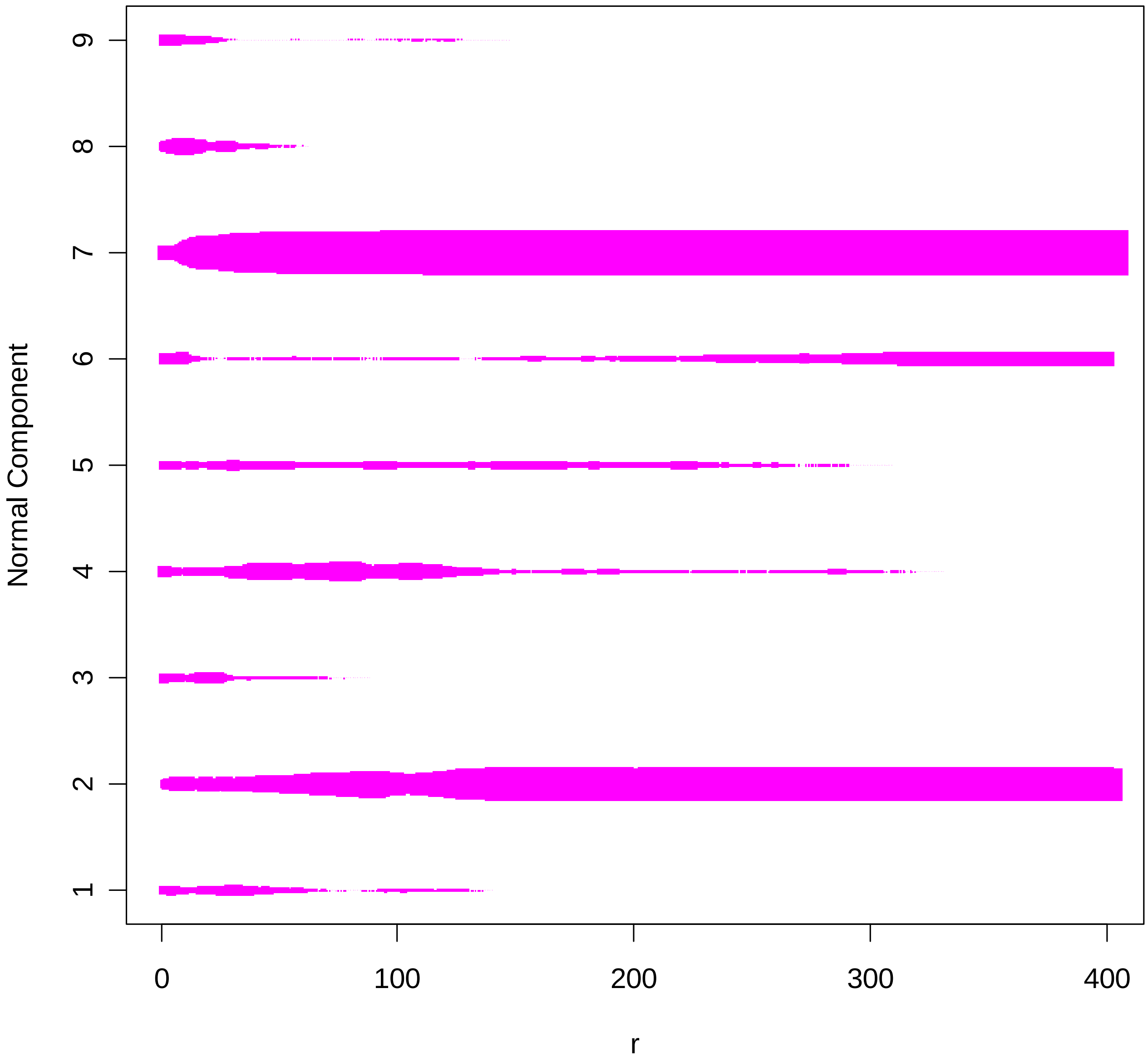


Figure 3.13
Posterior Draws of Marginal Densities:
Multivariate Normal Mixture Example

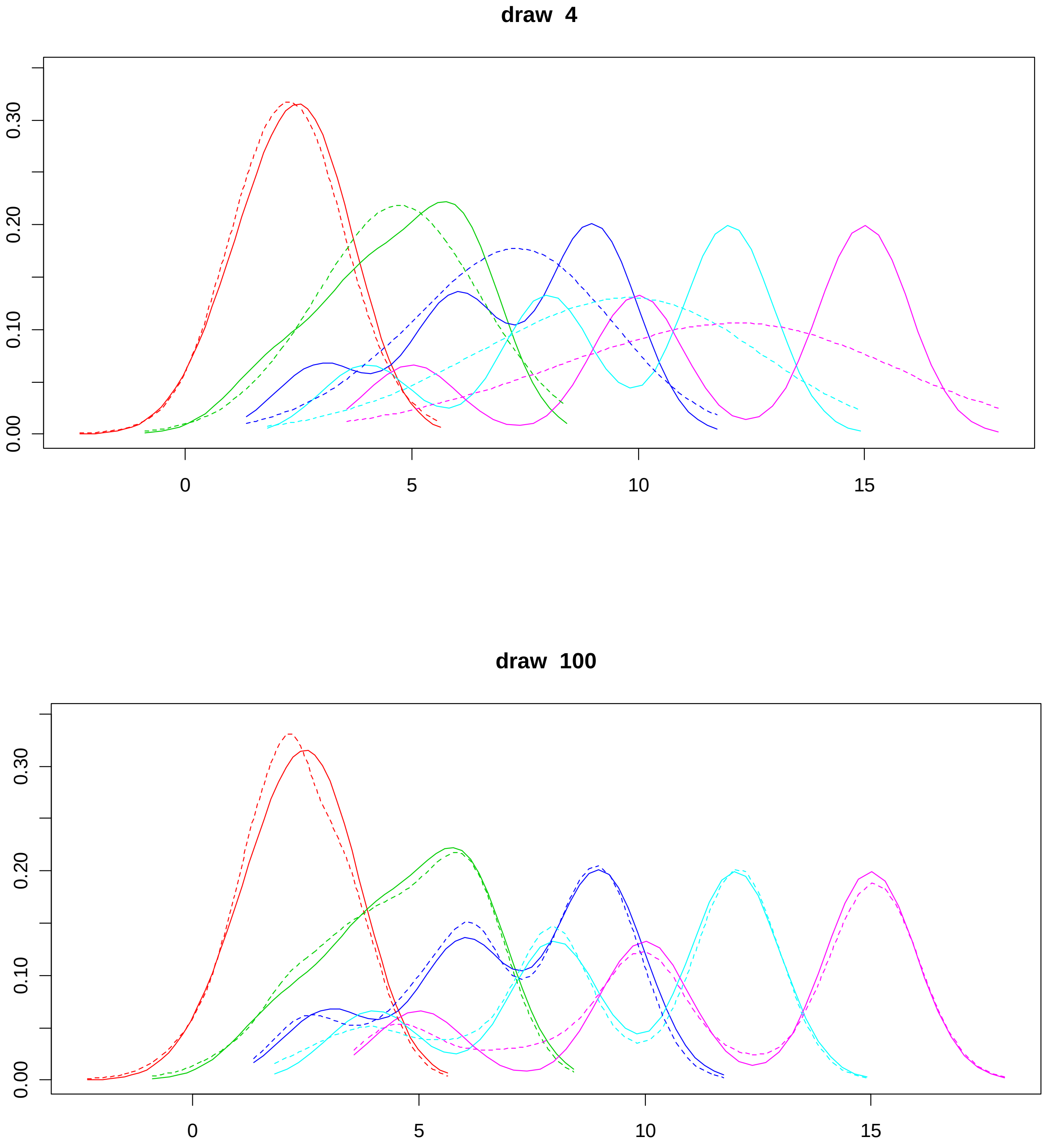


Figure 3.14
Optimal Scaling of the RW Metropolis for MNL Logit Example

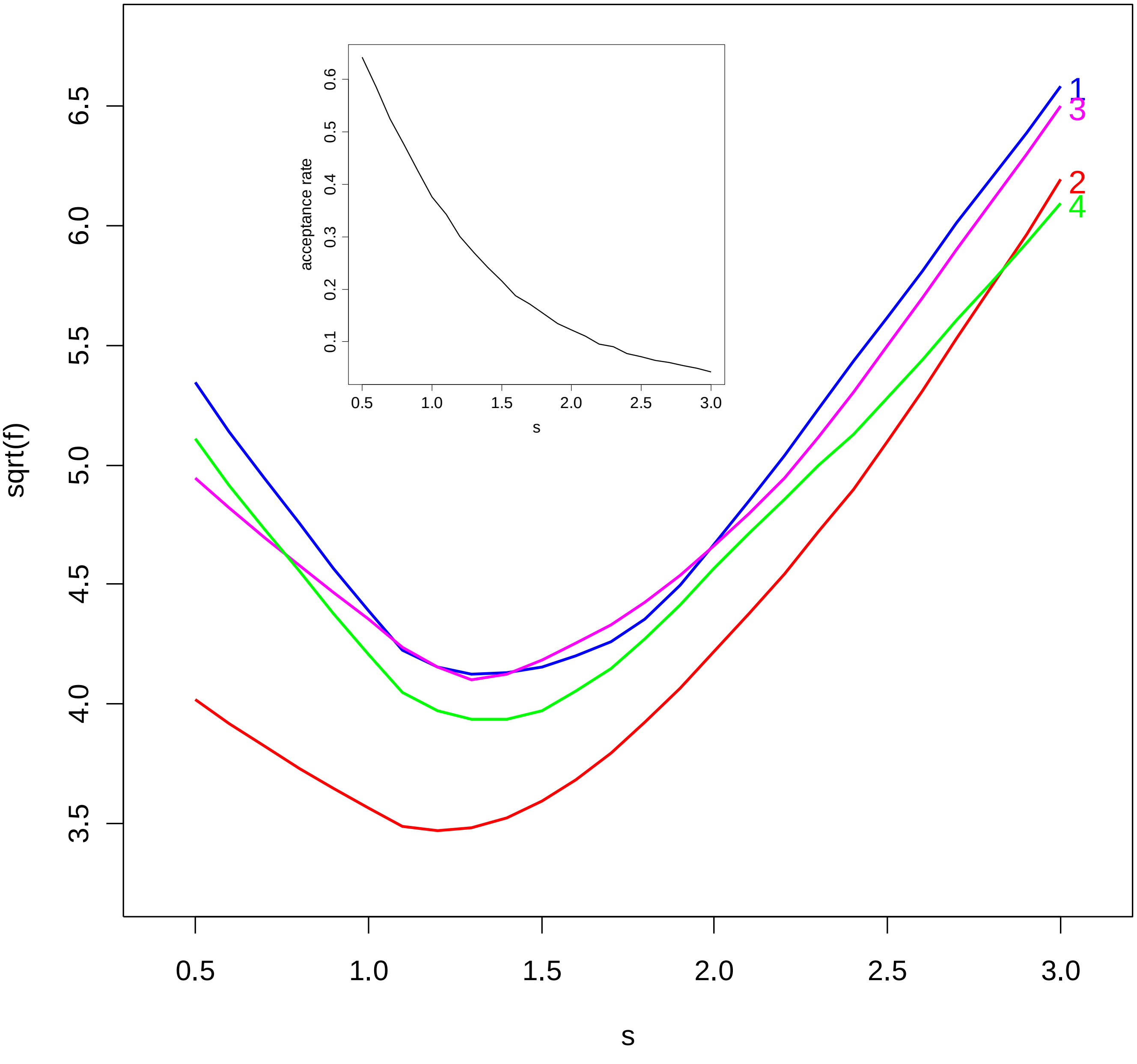
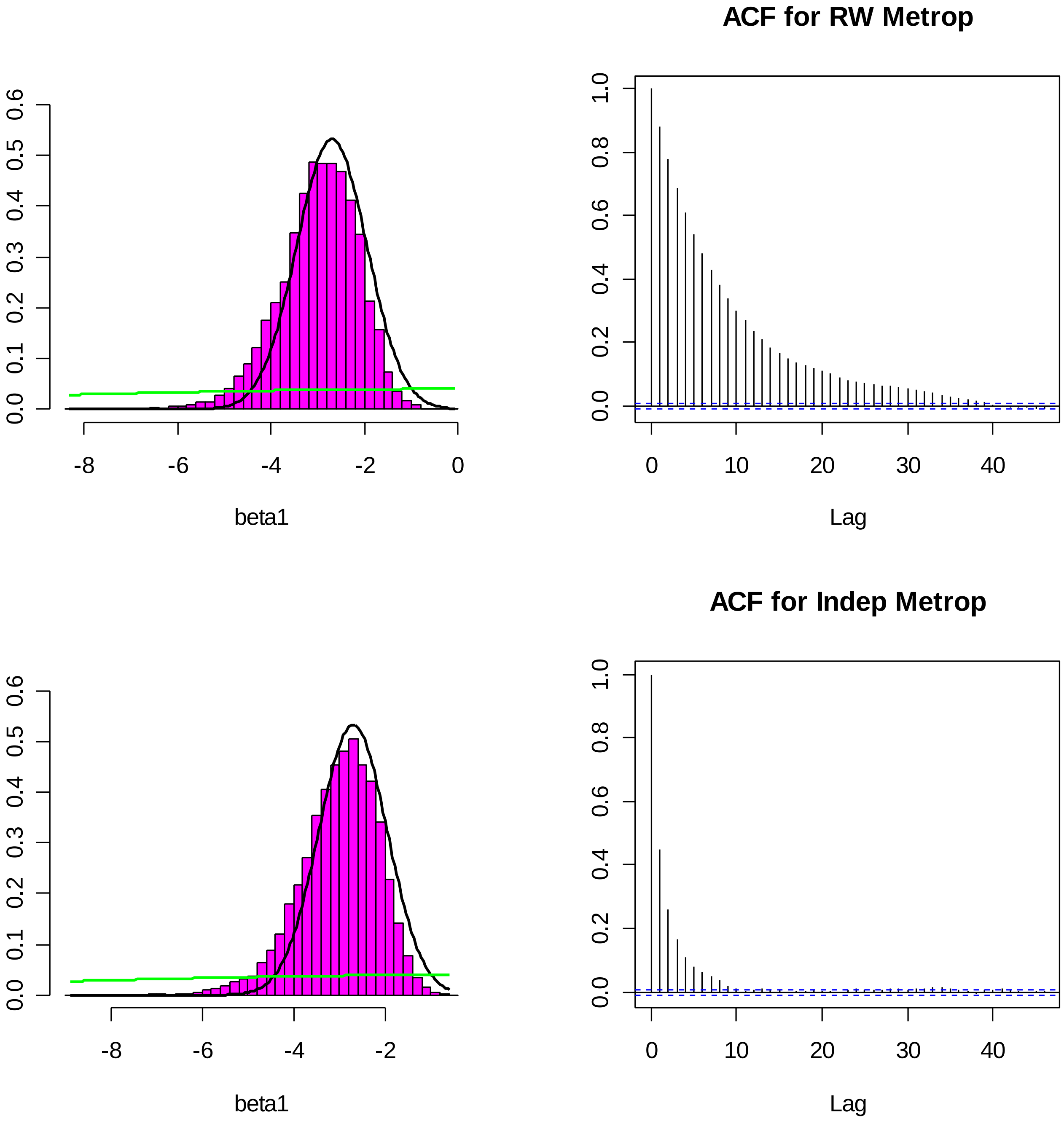


Figure 3.15
Comparison of RW and Independence Metropolis for MNL Example


ACF for RW Metrop
beta1
Lag
ACF for Indep Metrop
beta1
Lag