2
Markov Chain Monte Carlo Sampling

Recently, Monte Carlo (MC) based sampling methods for evaluating high-dimensional posterior integrals have been rapidly developing. Those sampling methods include MC importance sampling (Hammersley and Handscomb 1964; Ripley 1987; Geweke 1989; Wolpert 1991), Gibbs sampling (Geman and Geman 1984; Gelfand and Smith 1990), Hit-and-Run sampling (Smith 1984; Bélisle, Romeijn, and Smith 1993; Chen 1993; Chen and Schmeiser 1993 and 1996), Metropolis–Hastings sampling (Metropolis et al. 1953; Hastings 1970; Green 1995), and hybrid methods (e.g., Müller 1991; Tierney 1994; Berger and Chen 1993). A general discussion of the Gibbs sampler and other Markov chain Monte Carlo (MCMC) methods is given in the *Journal of the Royal Statistical Society, Series B* (1993), and an excellent roundtable discussion on the practical use of MCMC can be found in Kass et al. (1998). Other discussions or instances of the use of MCMC sampling can be found in Tanner and Wong (1987), Tanner (1996), Geyer (1992), Gelman and Rubin (1992), Gelfand, Smith, and Lee (1992), Gilks and Wild (1992), and many others. Further development of state-of-the-arts MCMC sampling techniques include the accelerated MCMC sampling of Liu and Sabatti (1998, 1999), Liu (1998), and Liu and Wu (1997), and the exact MCMC sampling of Green and Murdoch (1999). Comprehensive accounts of MCMC methods and their applications may also be found in Meyn and Tweedie (1993), Tanner (1996), Gilks, Richardson, and Spiegelhalter (1996), Robert and Casella (1999), and Gelfand and Smith (2000). The purpose of this chapter is to give a brief overview of several commonly used MCMC sampling algorithms as well as to present selectively several newly developed computational tools for MCMC sampling.

2.1 Gibbs Sampler

The Gibbs sampler may be one of the best known MCMC sampling algorithms in the Bayesian computational literature. As discussed in Besag and Green (1993), the Gibbs sampler is founded on the ideas of Grenander (1983), while the formal term is introduced by Geman and Geman (1984). The primary bibliographical landmark for Gibbs sampling in problems of Bayesian inference is Gelfand and Smith (1990). A similar idea termed as *data augmentation* is introduced by Tanner and Wong (1987). Casella and George (1992) provide an excellent tutorial on the Gibbs sampler.

Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)'$ be a p-dimensional vector of parameters and let $\pi(\boldsymbol{\theta}|D)$ be its posterior distribution given the data D. Then, the basic scheme of the Gibbs sampler is given as follows:

Gibbs Sampling Algorithm

Step 0. Choose an arbitrary starting point $\boldsymbol{\theta}_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{p,0})'$, and set $i = 0$.

Step 1. Generate $\boldsymbol{\theta}_{i+1} = (\theta_{1,i+1}, \theta_{2,i+1}, \ldots, \theta_{p,i+1})'$ as follows:

- Generate $\theta_{1,i+1} \sim \pi(\theta_1|\theta_{2,i}, \ldots, \theta_{p,i}, D)$;
- Generate $\theta_{2,i+1} \sim \pi(\theta_2|\theta_{1,i+1}, \theta_{3,i}, \ldots, \theta_{p,i}, D)$;
  ... ...
- Generate $\theta_{p,i+1} \sim \pi(\theta_p|\theta_{1,i+1}, \theta_{2,i+1}, \ldots, \theta_{p-1,i+1}, D)$.

Step 2. Set $i = i + 1$, and go to Step 1.

Thus each component of $\boldsymbol{\theta}$ is visited in the natural order and a cycle in this scheme requires generation of p random variates. Gelfand and Smith (1990) show that under certain regularity conditions, the vector sequence $\{\boldsymbol{\theta}_i,\ i = 1, 2, \ldots\}$ has a stationary distribution $\pi(\boldsymbol{\theta}|D)$. Schervish and Carlin (1992) provide a sufficient condition that guarantees geometric convergence. Other properties regarding geometric convergence are discussed in Roberts and Polson (1994). To illustrate the Gibbs sampler, we consider the following two simple examples:

Example 2.1. Bivariate normal model. The purpose of this example is to examine the exact correlation structure of the Markov chain induced by the Gibbs sampler. Assume that the posterior distribution $\pi(\boldsymbol{\theta}|D)$ is a bivariate normal distribution $N_2(\boldsymbol{\mu}, \Sigma)$ with

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \text{ and } \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

where μ_j, σ_j, $j = 1, 2$, and ρ are known. Then the Gibbs sampler requires sampling from

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

and

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

Let $\{\boldsymbol{\theta}_i = (\theta_{1,i}, \theta_{2,i})', \ i \geq 0\}$ denote the Markov chain induced by the Gibbs sampler for the above bivariate normal distribution. If we start from the stationary distribution, i.e., $\boldsymbol{\theta}_0 \sim N(\mu, \Sigma)$, then each of $\{\theta_{1,i}, \ i \geq 0\}$ and $\{\theta_{2,i}, \ i \geq 0\}$ is an AR(1) process.

To see this, let $\{z_{1,i}, z_{2,i}, \ i \geq 0\}$ be an i.i.d. $N(0,1)$ random variable sequence. Then the structure of the Gibbs sampler implies

$$\begin{aligned}
\theta_{1,0} &= \mu_1 + \sigma_1 z_{1,0}, \\
\theta_{2,0} &= \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,0} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,0},
\end{aligned}$$

and

$$\begin{aligned}
\theta_{1,i+1} &= \mu_1 + \rho\frac{\sigma_1}{\sigma_2}(\theta_{2,i} - \mu_2) + \sigma_1\sqrt{1-\rho^2}z_{1,i+1}, \\
\theta_{2,i+1} &= \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i+1} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,i+1},
\end{aligned} \tag{2.1.1}$$

for $i \geq 0$, Now, we consider the first component $\theta_{1,i+1}$. From (2.1.1), for $i \geq 0$,

$$\begin{aligned}
\theta_{1,i+1} =& \mu_1 + \rho\frac{\sigma_1}{\sigma_2}\left[\rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,i}\right] \\
&+ \sigma_1\sqrt{1-\rho^2}z_{1,i+1} \\
=& \mu_1 + \rho^2(\theta_{1,i} - \mu_1) + \rho\sigma_1\sqrt{1-\rho^2}z_{2,i} \\
&+ \sigma_1\sqrt{1-\rho^2}z_{1,i+1}.
\end{aligned} \tag{2.1.2}$$

Let $\psi = \rho^2$ and $\sigma_1^{*2} = \sigma_1^2(1-\rho^4)$. Let $\{z_i^*, \ i \geq 0\}$ denote an i.i.d. $N(0,1)$ random variable sequence. Since $z_{1,i}$ and $z_{2,i+1}$ are independently and identically distributed as $N(0,1)$, then we can rewrite (2.1.2) as

$$\theta_{1,0} = \mu_1 + \sigma_1 z_0^*, \tag{2.1.3}$$

$$\theta_{1,i+1} = \mu_1 + \psi(\theta_{1,i} - \mu_1) + \sigma_1^* z_{i+1}^* \quad \text{for } i \geq 0. \tag{2.1.4}$$

Thus, $\{\theta_{1,i}, \ i \geq 0\}$ is an AR(1) process with lag-one autocorrelation $\psi = \rho^2$. Similarly, $\{\theta_{2,i}, \ i \geq 0\}$ is also an AR(1) process with lag-one autocorrelation $\psi = \rho^2$. The only difference is that we use $\sigma_2^* = \sigma_2\sqrt{1-\rho^4}$ instead of σ_1^* in (2.1.4), and use μ_2 and σ_2 instead of μ_1 and σ_1 in (2.1.3).

Roberts and Sahu (1997) obtain a similar result for a general multivariate normal target distribution $\pi(\theta|D)$, that is, the Markov chain induced by the Gibbs sampler is a multivariate AR(1) process.

Example 2.2. Constrained multiple linear regression model. We consider a constrained multiple linear regression model given by (1.3.1) to model the New Zealand apple data described in Example 1.1. Let

$$\Omega = \{\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{10})' : 0 \leq \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{10}, \boldsymbol{\beta} \in R^{10}\} \quad (2.1.5)$$

denote the constraints given in (1.3.2). We take a joint prior for $(\boldsymbol{\beta}, \sigma^2)$ of the form

$$\pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}\pi(\beta_{10}|\mu_{10}, \sigma^2_{10}), \quad (2.1.6)$$

for $\sigma^2 > 0$ and $\boldsymbol{\beta} \in \Omega$, where $\pi(\beta_{10}|\mu_{10}, \sigma^2_{10})$ is a normal density with mean μ_{10} and variance σ^2_{10}. The modification of the usual flat noninformative prior to include the informative distribution on β_{10} is necessary to prevent too much weight being given to the unconstrained and therefore unbounded parameter β_{10}. Chen and Deely (1996) specify $\mu_{10} = 0.998$, and $\sigma^2_{10} = 0.089$ by using method-of-moments, a well–known type of empirical Bayes estimation, from the data on growers with mature trees only. Using (2.1.6), the posterior distribution for $(\boldsymbol{\beta}, \sigma^2)$ based on the New Zealand apple data D is given by

$$\pi(\boldsymbol{\beta}, \sigma^2|D) = \frac{\exp\{-(\beta_{10} - \mu_{10})^2/2\sigma^2_{10}\}}{c(D)(\sigma^2)^{(n+1)/2}} \times \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{10} x_{ij}\beta_j\right)^2\right\}, \quad (2.1.7)$$

for $\sigma^2 > 0$ and $\boldsymbol{\beta} \in \Omega$, where y_i is the total number of cartons of fruit produced and x_{ij} = number of trees at age $j+1$ for $j = 1, 2, \ldots, 10$ for the i^{th} grower, $c(D)$ is the normalizing constant, and $n = 207$ denotes the sample size. Due to the constraints, the analytical evaluation of posterior quantities such as the posterior mean and posterior standard deviation of β_j does not appear possible. However, the implementation of the Gibbs sampler for sampling from the posterior (2.1.7) is straightforward. More specifically, we run the Gibbs sampler by taking

$$\beta_j|\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots, \beta_{10}, \sigma^2, D \sim N(\theta_j, \delta_j^2) \quad (2.1.8)$$

subject to $\beta_{j-1} \leq \beta_j \leq \beta_{j+1}$ $(\beta_0 = 0)$ for $j = 1, 2, \ldots, 9$,

$$\beta_{10}|\beta_1, \ldots, \beta_9, \sigma^2, D \sim N(\psi\theta_{10} + (1-\psi)\mu_{10}, (1-\psi)\sigma^2_{10}) \quad (2.1.9)$$

subject to $\beta_{10} \geq \beta_9$ and

$$\sigma^2|\boldsymbol{\beta}, D \quad \sim \quad \mathcal{IG}\left(\frac{n}{2}, \frac{\sum_{i=1}^{n}(y_i - \sum_{j=1}^{10} x_{ij}\beta_j)^2}{2}\right), \tag{2.1.10}$$

where in (2.1.8) and (2.1.9), $\psi = \sigma_{10}^2/(\sigma_{10}^2 + \delta_{10}^2)$,

$$\theta_j = \left(\sum_{i=1}^{n} x_{ij}^2\right)^{-1} \left[\sum_{i=1}^{n}\left(y_i - \sum_{l \neq j} x_{il}\beta_l\right) x_{ij}\right], \tag{2.1.11}$$

and

$$\delta_j^2 = \left(\sum_{i=1}^{n} x_{ij}^2\right)^{-1} \sigma^2 \tag{2.1.12}$$

for $j = 1, \ldots, 10$, and $\mathcal{IG}(\xi, \eta)$ denotes the inverse gamma distribution with parameters (ξ, η), whose density is given by

$$\pi(\sigma^2|\xi, \eta) \;\propto\; (\sigma^2)^{-(\xi+1)} e^{-\eta/\sigma^2}.$$

2.2 Metropolis–Hastings Algorithm

The Metropolis–Hastings algorithm is developed by Metropolis et al. (1953) and subsequently generalized by Hastings (1970). Tierney (1994) gives a comprehensive theoretical exposition of this algorithm, and Chib and Greenberg (1995) provide an excellent tutorial on this topic.

Let $q(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ be a proposal density, which is also termed as a *candidate-generating density* by Chib and Greenberg (1995), such that

$$\int q(\boldsymbol{\theta}, \boldsymbol{\vartheta})\, d\boldsymbol{\vartheta} = 1.$$

Also let $U(0, 1)$ denote the uniform distribution over $(0, 1)$. Then, a general version of the Metropolis–Hastings algorithm for sampling from the posterior distribution $\pi(\boldsymbol{\theta}|D)$ can be described as follows:

Metropolis–Hastings Algorithm

Step 0. Choose an arbitrary starting point $\boldsymbol{\theta}_0$ and set $i = 0$.

Step 1. Generate a candidate point $\boldsymbol{\theta}^*$ from $q(\boldsymbol{\theta}_i, \cdot)$ and u from $U(0, 1)$.

Step 2. Set $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}^*$ if $u \leq a(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)$ and $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i$ otherwise, where the acceptance probability is given by

$$a(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \min\left\{\frac{\pi(\boldsymbol{\vartheta}|D)q(\boldsymbol{\vartheta}, \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|D)q(\boldsymbol{\theta}, \boldsymbol{\vartheta})}, 1\right\}. \tag{2.2.1}$$

Step 3. Set $i = i + 1$, and go to Step 1.

The above algorithm is very general. When $q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q(\boldsymbol{\vartheta})$, the Metropolis–Hastings algorithm reduces to the *independence chain* Metropolis algorithm (see Tierney 1994). More interestingly, the Gibbs sampler is obtained as a special case of the Metropolis–Hastings algorithm by choosing an appropriate $q(\boldsymbol{\theta}, \boldsymbol{\vartheta})$. This relationship is first pointed out by Gelman (1992) and further elaborated on by Chib and Greenberg (1995).

Another family of proposal densities is given by the form $q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q_1(\boldsymbol{\vartheta} - \boldsymbol{\theta})$, where $q_1(\cdot)$ is a multivariate density (see Müller 1991). The candidate $\boldsymbol{\theta}^*$ is thus drawn according to the process $\boldsymbol{\theta}^* = \boldsymbol{\theta} + \boldsymbol{\omega}$, where $\boldsymbol{\omega}$ is called the increment random variable and follows the distribution q_1. Because the candidate is equal to the current value plus noise, Chib and Greenberg (1995) call this case a *random walk chain.* Many other algorithms such as the Hit-and-Run algorithm and dynamic weighting algorithm, which will be presented later in this chapter, are also special cases of this general algorithm.

The performance of a Metropolis–Hastings algorithm depends on the choice of a proposal density q. As discussed in Chib and Greenberg (1995), the spread of the proposal density q affects the behavior of the chain in at least two dimensions: one is the "acceptance rate" (the percentage of times a move to a new point is made), and the other is the region of the sample space that is covered by the chain. If the spread is extremely large, some of the generated candidates will have a low probability of being accepted. On the other hand, if the spread is chosen too small, the chain will take longer to traverse the support of the density. Both of these situations are likely to be reflected in high autocorrelations across sample values. In the context of q_1 (the random walk proposal density), Roberts, Gelman, and Gilks (1997) show that if the target and proposal densities are normal, then the scale of the latter should be tuned so that the acceptance rate is approximately 0.45 in one-dimensional problems and approximately 0.23 as the number of dimensions approaches infinity, with the optimal acceptance rate being around 0.25 in six dimensions. For the *independence chain*, in which we take $q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q(\boldsymbol{\vartheta})$, it is important to ensure that the tails of the proposal density $q(\boldsymbol{\vartheta})$ dominate those of the target density $\pi(\boldsymbol{\theta}|D)$, which is similar to a requirement on the importance sampling function in Monte Carlo integration with importance sampling (Geweke 1989).

To illustrate the Metropolis–Hastings algorithm, we consider a problem of sampling a correlation coefficient ρ from its posterior distribution.

Example 2.3. An algorithm for sampling a correlation ρ. Assume that $D = \{\boldsymbol{y}_i = (y_{1i}, y_{2i})', \ i = 1, 2, \ldots, n\}$ is a random sample from a

bivariate normal distribution $N_2(0, \Sigma)$, where

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Assuming a uniform prior $U(-1, 1)$ for ρ, the posterior density for ρ is given by

$$\pi(\rho|D) \propto (1-\rho^2)^{-n/2} \exp\left\{ -\frac{1}{2(1-\rho^2)}(S_{11} - 2\rho S_{12} + S_{22}) \right\}, \quad (2.2.2)$$

where $-1 < \rho < 1$, and $S_{rs} = \sum_{i=1}^{n} y_{ri} y_{si}$ for $r, s = 1, 2$. Generating ρ from (2.2.2) is not trivial since $\pi(\rho|D)$ is not log-concave. Therefore, we consider the following Metropolis–Hastings algorithm with a "de-constraint" transformation to sample ρ. Since $-1 < \rho < 1$, we let

$$\rho = \frac{-1 + e^{\xi}}{1 + e^{\xi}}, \quad -\infty < \xi < \infty. \quad (2.2.3)$$

Then

$$\pi(\xi|D) = \pi(\rho|D) \frac{2e^{\xi}}{(1+e^{\xi})^2}.$$

Instead of directly sampling ρ, we generate ξ by choosing a normal proposal $N(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$, where $\hat{\xi}$ is a maximizer of the logarithm of $\pi(\xi|D)$, which can be obtained by, for example, the standard Newton–Raphson algorithm or the Nelder–Mead algorithm implemented by O'Neill (1971), and $\hat{\sigma}^2_{\hat{\xi}}$ is minus the inverse of the second derivative of $\log \pi(\xi|D)$ evaluated at $\xi = \hat{\xi}$, that is,

$$\hat{\sigma}^{-2}_{\hat{\xi}} = - \left. \frac{d^2 \log \pi(\xi|D)}{d\xi^2} \right|_{\xi=\hat{\xi}}.$$

The algorithm to generate ξ operates as follows:

Step 1. Let ξ be the current value.

Step 2. Generate a proposal value ξ^* from $N(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$.

Step 3. A move from ξ to ξ^* is made with probability

$$\min \left\{ \frac{\pi(\xi^*|D)\phi\left(\frac{\xi - \hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)}{\pi(\xi|D)\phi\left(\frac{\xi^* - \hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)}, 1 \right\}, \quad (2.2.4)$$

where ϕ is the standard normal probability density function.

After we obtain ξ, we compute ρ by using (2.2.3).

Since the above algorithm does not use a random walk proposal density, the optimal acceptance rate, 0.23, of Roberts, Gelman, and Gilks (1997)

cannot be applied here. A detailed study of how this algorithm performs is thus left as an exercise. The above algorithm can also be extended to the cases where $\pi(\rho|D)$ is a conditional posterior distribution that depends on other parameters. For example, the conditional posterior distribution for ρ may be written as $\pi(\rho|\boldsymbol{\theta}, D)$. Then, the Metropolis–Hastings algorithm to sample from $\pi(\rho|\boldsymbol{\theta}, D)$ proceeds in a similar way. The idea of a normal proposal that is matched to the conditional posterior appears for the first time in Chib and Greenberg (1994). A nice feature of this extension is that the normal proposal density for this more general case becomes adaptive since it depends on the values of the other parameters from the current and previous iterations. This semiautomatic updating feature makes the proposal density closer to the true conditional posterior, which may lead to a more efficient Metropolis–Hastings algorithm.

2.3 Hit-and-Run Algorithm

The Hit-and-Run (H&R) algorithm is a special case of the Metropolis–Hastings algorithm. Its original form is proposed independently by Boneh and Golan (1979) and Smith (1980) for generating points uniformly distributed over bounded regions in mathematical programming problems. Smith (1984) calls the H&R a "Mixing Algorithm" and he then proves the convergence of the algorithm. This algorithm has not been studied for about 10 years until Bélisle, Romeijn, and Smith (1993) propose a more general form of the H&R algorithm that generates a sample of points from an arbitrary continuous target distribution. However, Bélisle, Romeijn, and Smith (1993) prove the convergence assuming that the target density is bounded and has bounded support. Chen and Schmeiser (1996) further generalize the H&R algorithm to a general target density for evaluating multidimensional integrals and Chen and Schmeiser (1993) also consider the performance of H&R compared to the Gibbs sampler. In the context of Bayesian computation, Berger and Chen (1993) use the H&R for sampling from a multinomial distribution with a constrained parameter space; Yang and Berger (1994) apply the H&R algorithm for estimation of a covariance matrix using the reference prior; and Yang and Chen (1995) employ the H&R algorithm with parameter transformations for Bayesian analysis of random coefficient regression models using noninformative priors. A slightly different but related algorithm termed *adaptive direction sampling* can be found in Gilks, Roberts, and George (1994) and Roberts and Gilks (1994).

Assume that the posterior distribution $\pi(\boldsymbol{\theta}|D)$ has support Ω. Then, the general H&R algorithm, requiring a distribution for the direction, a density g_i for the signed distance, and an acceptance probability a_i, can be stated as follows:

Hit-and-Run Algorithm

Step 0. Choose an arbitrary starting point $\boldsymbol{\theta}_0$ and set $i = 0$.

Step 1. Generate a direction $\boldsymbol{d}_i$ from a distribution on the surface of the unit sphere.

Step 2. Find the set $\Omega_i = \Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i) = \{\lambda \in R | \boldsymbol{\theta}_i + \lambda \boldsymbol{d}_i \in \Omega\}$.

Step 3. Generate a signed distance λ_i from density $g_i(\lambda | \, \boldsymbol{d}_i, \boldsymbol{\theta}_i)$, where $\lambda_i \in \Omega_i$.

Step 4. Set $\boldsymbol{\theta}^* = \boldsymbol{\theta}_i + \lambda_i \boldsymbol{d}_i$. Then set

$$\boldsymbol{\theta}_{i+1} = \begin{cases} \boldsymbol{\theta}^*, & \text{with the probability } \ a_i(\boldsymbol{\theta}^* | \boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i, & \text{otherwise.} \end{cases} \tag{2.3.1}$$

Step 5. Set $i = i + 1$, and go to Step 1.

Chen and Schmeiser (1996) discuss various choices for the distributions of $\boldsymbol{d}_i$, the densities g_i, and the probabilities a_i. Let the distribution of the direction $\boldsymbol{d}_i$, as used in Step 2 of H&R, have density $r(\boldsymbol{d}_i)$, with the surface of the unit sphere as its support. Then, assume that:

(i) for any density $g_i(\lambda | \boldsymbol{d}_i, \boldsymbol{\theta}_i)$ in Step 3, $g_i(\lambda | \boldsymbol{d}_i, \boldsymbol{\theta}_i) > 0$ and

$$g_i(-\lambda | \, -\boldsymbol{d}_i, \boldsymbol{\theta}_i) = g_i(\lambda | \boldsymbol{d}_i, \boldsymbol{\theta}_i);$$

(ii) for the distribution of the direction, $r(\boldsymbol{d}_i) > 0$;

(iii) for any a_i in Step 4, $0 < a_i(\boldsymbol{\theta}^* | \boldsymbol{\theta}_i) \leq 1$; and

(iv) for any $\boldsymbol{\theta}, \boldsymbol{\theta}^* \in \Omega$

$$g_i\left(\| \, \boldsymbol{\theta} - \boldsymbol{\theta}^* \, \| \, \left| \frac{\boldsymbol{\theta}^* - \boldsymbol{\theta}}{\| \, \boldsymbol{\theta}^* - \boldsymbol{\theta} \, \|}, \theta \right. \right) \cdot a_i(\boldsymbol{\theta}^* | \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | D)$$
$$= g_i\left(\| \, \boldsymbol{\theta}^* - \boldsymbol{\theta} \, \| \, \left| \frac{\boldsymbol{\theta} - \boldsymbol{\theta}^*}{\| \, \boldsymbol{\theta} - \boldsymbol{\theta}^* \, \|}, \boldsymbol{\theta}^* \right. \right) \cdot a_i(\boldsymbol{\theta} | \boldsymbol{\theta}^*) \pi(\boldsymbol{\theta}^* | D).$$

Under the assumptions above, the Markov chain $\{\boldsymbol{\theta}_i, \ i = 0, 1, 2, \ldots\}$ converges to its stationary distribution $\pi(\boldsymbol{\theta} | D)$.

The most common choice of $r(\boldsymbol{d}_i)$ is a uniform distribution on the surface of the unit sphere. Common choices of g_i and a_i are given as follows:

Choice I:

$$g_i^{\mathrm{I}}(\lambda | \boldsymbol{d}_i, \boldsymbol{\theta}_i) = \frac{\pi(\boldsymbol{\theta}_i + \lambda \boldsymbol{d}_i | D)}{\int_{\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i)} \pi(\boldsymbol{\theta}_i + u \boldsymbol{d}_i | D) \, du} \quad \text{for } \lambda \in \Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i),$$

and

$$a_i^{\mathrm{I}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = a_i^{\mathrm{I}}(\boldsymbol{\theta}_i|\boldsymbol{\theta}^*), \quad 0 < a^{\mathrm{I}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) \leq 1 \text{ for all } \boldsymbol{\theta}_i, \boldsymbol{\theta}^* \in \Omega.$$

Typically $a_i^{\mathrm{I}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = 1$.

Choice II:

Choose $g_i(\lambda|\boldsymbol{d}_i, \boldsymbol{\theta}_i)$ to be one of the following:

(a) If Ω is bounded, then

$$g_i^{\mathrm{II}}(\lambda|\boldsymbol{d}_i, \boldsymbol{\theta}_i) = \frac{1}{m(\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i))} \quad \text{for } \lambda \in \Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i),$$

where m denotes Lebesgue measure.

(b) If Ω is unbounded, then choose $g_i^{\mathrm{II}}(\lambda|\boldsymbol{d}_i, \boldsymbol{\theta}_i)$ to be a symmetric-about-zero, continuous distribution with unbounded support $\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i)$ and shape depending only on $\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i)$. For example, g_i^{II} can be a normal distribution, Cauchy distribution, or double-exponential distribution with location parameter zero and scale parameter depending only on $\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i)$.

Independent of the choice (a) or (b), choose $a_i(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$ to be either:

(c) Barker's method (Barker 1965)

$$a_i^{\mathrm{II}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = \frac{\pi(\boldsymbol{\theta}^*|D)}{\pi(\boldsymbol{\theta}_i|D) + \pi(\boldsymbol{\theta}^*|D)}.$$

or

(d) Metropolis's method

$$a_i^{\mathrm{II}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*|D)}{\pi(\boldsymbol{\theta}_i|D)}\right).$$

Choice III:

Choose $g_i^{\mathrm{III}}(\lambda|\boldsymbol{d}_i, \boldsymbol{\theta}_i) = g_i(\boldsymbol{\theta}_i + \lambda \boldsymbol{d}_i)$, where g_i depends only on $\Omega_i(\boldsymbol{d}_i, \boldsymbol{\theta}_i)$, and

$$a_i^{\mathrm{III}}(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = \min\{\omega(\boldsymbol{\theta}_i + \lambda \boldsymbol{d}_i)/\omega(\boldsymbol{\theta}_i), 1\},$$

where $\omega(\boldsymbol{\theta}_i) = \pi(\boldsymbol{\theta}_i|D)/g_i(\boldsymbol{\theta}_i)$.

These choices are motivated by Hastings (1970). For a given g_i in Choice III, the results of Peskun (1973) imply that when Ω is a finite set, the choice of a_i^{III} is optimal in the sense of minimizing the asymptotic variance of the sample average $(1/n)\sum_{i=1}^n h(\boldsymbol{\theta}_i)$, where $h(\cdot)$ is a real function of $\boldsymbol{\theta}$

satisfying

$$\int_{R^p} |h(\boldsymbol{\theta})|\pi(\boldsymbol{\theta}|D)\ d\boldsymbol{\theta} < \infty.$$

With Choice I, Kaufman and Smith (1998) develop an optimal direction choice algorithm for H&R and prove that there exists a unique optimal direction choice distribution for $r(\cdot)$. The other theoretical properties of H&R can be found in Bélisle, Romeijn, and Smith (1993), and Chen and Schmeiser (1993, 1996). Regarding applications of H&R to Bayesian computation, Berger (1993) comments that

> *"This method is particularly useful when $\boldsymbol{\theta}$ has a sharply constrained parameter space."*

To illustrate the H&R algorithm, we revisit the constrained multiple linear regression model discussed in Section 2.1.

Example 2.4. Constrained multiple linear regression model (Example 2.2 continued). Instead of using the Gibbs sampler to sample $(\boldsymbol{\beta}, \sigma^2)$ from $\pi(\boldsymbol{\beta}, \sigma^2|D)$ given in (2.1.7), we use the H&R algorithm. All eleven dimensions are sampled within a Gibbs sampling framework, with the ten $\boldsymbol{\beta}$ dimensions sampled with H&R and σ^2 sampled from its known conditional gamma density in the Gibbs step. For illustrative purposes, we state the H&R logic for sampling $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{10})'$ from its conditional posterior distribution for a given value of σ^2 and D:

Step 0. Choose a starting point $\boldsymbol{\beta}_0 \in \Omega$ and set $i = 0$.

Step 1. Generate a uniformly distributed unit-length direction $\boldsymbol{d}_i = (d_{1,i}, d_{2,i}, \ldots, d_{10,i})'$.

Step 2. Find the set $\Omega_i = (R_1^i, R_2^i)$, where

$$R_1^i = \inf_{\lambda}\{\lambda : \boldsymbol{\beta}_i + \lambda \boldsymbol{d}_i \in \Omega\} \text{ and } R_2^i = \sup_{\lambda}\{\lambda : \boldsymbol{\beta}_i + \lambda \boldsymbol{d}_i \in \Omega\}.$$

Step 3. Generate a signed distance λ_i from the density

$$\pi_i(\lambda) = \frac{\pi(\boldsymbol{\beta}_i + \lambda \boldsymbol{d}_i, \sigma^2|D)}{\int_{R_1^i}^{R_2^i} \pi(\boldsymbol{\beta}_i + u\boldsymbol{d}_i, \sigma^2|D)\ du}, \quad \lambda \in (R_1^i, R_2^i). \tag{2.3.2}$$

Step 4. Set $\boldsymbol{\beta}_{i+1} = \boldsymbol{\beta}_i + \lambda_i \boldsymbol{d}_i$.

Step 5. Set $i = i + 1$ and go to Step 1.

Here we use the probability $a_i = 1$. Sampling in each step is straightforward. A random unit-length direction $\boldsymbol{d}_i$ can be generated in Step 2 by

independently generating $z_l \sim N(0,1)$ and setting

$$d_{l,i} = z_l \left(\sum_{j=1}^{10} z_j^2 \right)^{-1/2}$$

for $l = 1, 2, ..., 10$; see, for example, Devroye (1986). The density given in (2.3.2) is a truncated normal probability density function, where the mean and variance are easy-to-compute functions of σ^2, $\boldsymbol{\beta}_i$, $\boldsymbol{d}_i$, and D. Computationally, the H&R algorithm is slightly more efficient than the usual (one-coordinate-at-a-time) Gibbs sampler. Implementation difficulty of the two sampling algorithms is similar.

2.4 Multiple-Try Metropolis Algorithm

Liu, Liang, and Wong (1998a) propose a novel algorithm, called the Multiple-Try Metropolis (MTM) algorithm. The algorithm proceeds as follows. Let $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ be a proposal transition density function, which may or may not be symmetric. A requirement for $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ is that $T(\boldsymbol{\theta}, \boldsymbol{\vartheta}) > 0$ if and only if $T(\boldsymbol{\vartheta}, \boldsymbol{\theta}) > 0$. Furthermore, define

$$w(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \pi(\boldsymbol{\theta}|D) T(\boldsymbol{\theta}, \boldsymbol{\vartheta}) \lambda(\boldsymbol{\theta}, \boldsymbol{\vartheta}), \tag{2.4.1}$$

where $\lambda(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ is a nonnegative symmetric function in $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ so that $\lambda(\boldsymbol{\theta}, \boldsymbol{\vartheta}) > 0$ whenever $T(\boldsymbol{\theta}, \boldsymbol{\vartheta}) > 0$. Suppose the current state is $\boldsymbol{\theta}_i$. In an MTM transition, the next state is generated as follows:

Multiple-Try Metropolis

Step 1. Generate k trials $\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \ldots, \boldsymbol{\vartheta}_k$ from the proposal distribution $T(\boldsymbol{\theta}_i, \boldsymbol{\vartheta})$. Compute $w(\boldsymbol{\vartheta}_j, \boldsymbol{\theta}_i)$ for $j = 1, 2, \ldots, k$.

Step 2. Select $\boldsymbol{\vartheta}_l$ among the $\boldsymbol{\vartheta}_j$'s with probability proportional to $w(\boldsymbol{\vartheta}_j, \boldsymbol{\theta}_i)$, $j = 1, 2, \ldots, k$. Then draw $\boldsymbol{\vartheta}_1^*, \boldsymbol{\vartheta}_2^*, \ldots, \boldsymbol{\vartheta}_{k-1}^*$ from the distribution $T(\boldsymbol{\vartheta}_l, \boldsymbol{\vartheta}^*)$, and let $\boldsymbol{\vartheta}_k^* = \boldsymbol{\theta}_i$.

Step 3. Generate u from $U(0,1)$. Set $\boldsymbol{\theta}_{i+1} = \boldsymbol{\vartheta}_l$ if $u \leq a$ and $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i$ otherwise, where the acceptance probability is given by

$$a = \min \left\{ 1, \frac{w(\boldsymbol{\vartheta}_1, \boldsymbol{\theta}_i) + w(\boldsymbol{\vartheta}_2, \boldsymbol{\theta}_i) + \cdots + w(\boldsymbol{\vartheta}_k, \boldsymbol{\theta}_i)}{w(\boldsymbol{\vartheta}_1^*, \boldsymbol{\vartheta}_l) + w(\boldsymbol{\vartheta}_2^*, \boldsymbol{\vartheta}_l) + \cdots + w(\boldsymbol{\vartheta}_k^*, \boldsymbol{\vartheta}_l)} \right\}.$$

Liu, Liang, and Wong (1998a) show that the MTM transition rule satisfies the detailed balance, and hence, induces a reversible MC with $\pi(\boldsymbol{\theta}|D)$ as its equilibrium distribution. They also present several choices of $\lambda(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ in (2.4.1). When $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ is symmetric and $\lambda(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = [T(\boldsymbol{\theta}, \boldsymbol{\vartheta})]^{-1}$, the MTM

algorithm reduces to the method of "orientation biased-Monte Carlo" described in Frenkel and Smit (1996), where they provide a specialized proof in the context of simulating molecular structures of materials. As discussed in Liu, Liang, and Wong (1998a), the MTM algorithm is more advantageous, since it allows one to explore more thoroughly the "neighboring region" defined by $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$, and it is particularly useful when one identifies certain directions of interest but has difficulty implementing a Gibbs sampling type move due to unfavorable conditional distributions. Liu, Liang, and Wong (1998a) also propose several variations of the MTM algorithm. These include a conjugate-gradient MC algorithm, a random-ray algorithm, and a Griddy–Gibbs MTM, which are closely related to the adaptive direction sampling algorithm of Gilks, Roberts, and George (1994) and Roberts and Gilks (1994), the H&R algorithm of Chen and Schmeiser (1993, 1996), and the Griddy–Gibbs algorithm of Ritter and Tanner (1992). For illustrative purposes, we briefly describe the random-ray algorithm as follows. Suppose the current state is $\boldsymbol{\theta}_i$. The random-ray algorithm executes the following update:

- Randomly generate a unit-length direction $\boldsymbol{d}$.
- Draw $\boldsymbol{\vartheta}_1$, $\boldsymbol{\vartheta}_2$, $\ldots$, $\boldsymbol{\vartheta}_k$ from the proposal transition $T_{\boldsymbol{d}}(\boldsymbol{\theta}_i, \boldsymbol{\vartheta})$ along the direction $\boldsymbol{d}$. One possible way to do this is to generate a random sample $\{r_1, r_2, \ldots, r_k\}$ from $N(0, \sigma^2)$, where σ^2 can be chosen large, and set $\boldsymbol{\vartheta}_k = \boldsymbol{\theta}_i + r_j \boldsymbol{d}$. Another approach is to generate $r_j \sim U[-\sigma, \sigma]$.
- Conduct the other MTM steps as described in the Multiple-Try Metropolis algorithm.

The implementational details for the other variations can be found in Liu, Liang, and Wong (1998a), and are omitted here for brevity.

2.5 Grouping, Collapsing, and Reparameterizations

In this section, we discuss several useful tools to improve convergence of MCMC sampling. In particular, we focus on the grouped and collapsed Gibbs techniques of Liu (1994) and Liu, Wong, and Kong (1994), and the hierarchical centering method of Gelfand, Sahu, and Carlin (1995, 1996).

2.5.1 Grouped and Collapsed Gibbs

Liu (1994) proposes a method of "grouping" and "collapsing" when using the Gibbs sampler in which he shows that both grouping and collapsing are beneficial based on operator theory. To illustrate his idea, we consider a three-dimensional posterior distribution $\pi(\boldsymbol{\theta}|D)$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)'$.

Liu (1994) considers the following three variations of the Gibbs sampler to sample from $\pi(\boldsymbol{\theta}|D)$:

Algorithm 1: *Standard (Original) Gibbs Sampler*

The standard Gibbs sampler requires drawing:

(i) $\theta_1 \sim \pi(\theta_1|\theta_2, \theta_3, D)$;

(ii) $\theta_2 \sim \pi(\theta_2|\theta_1, \theta_3, D)$;

(iii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

Algorithm 2: *Grouped Gibbs Sampler*

The grouped Gibbs sampler requires drawing:

(i) $(\theta_1, \theta_2) \sim \pi(\theta_1, \theta_2|\theta_3, D)$;

(ii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

In Algorithm 2, we first group (θ_1, θ_2) together and then simultaneously draw (θ_1, θ_2) from their joint conditional posterior distribution $\pi(\theta_1, \theta_2|\theta_3, D)$.

Algorithm 3: *Collapsed Gibbs Sampler*

The collapsed Gibbs sampler requires drawing:

(i) $(\theta_1, \theta_2) \sim \pi(\theta_1, \theta_2|D)$;

(ii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

The main difference between Algorithms 2 and 3 is the implementation of step (i). In particular, the collapsed Gibbs draws (θ_1, θ_2) from their marginal posterior distribution instead of the conditional posterior distribution as in Algorithm 2. Liu (1994) also mentions that if one uses a "mini-Gibbs" to draw (θ_1, θ_2) in step (i), that is, to sample $\theta_1 \sim \pi(\theta_1|\theta_2, D)$ and then $\theta_2 \sim \pi(\theta_2|\theta_1, D)$, the collapsed Gibbs requires that the chain from the mini-Gibbs sampler converges before step (ii). In practice, it may be difficult or expensive to directly draw (θ_1, θ_2) jointly from $\pi(\theta_1, \theta_2|D)$. In this case, we consider the following modified version of the collapsed Gibbs sampler:

Algorithm 3(a): *Modified Collapsed Gibbs Sampler*

The modified collapsed Gibbs sampler is similar to the original version by changing step (i) to:

(ia) $\theta_1 \sim \pi(\theta_1|\theta_2, D)$;

(ib) $\theta_2 \sim \pi(\theta_2|\theta_1, D)$.

We can show that the modified Gibbs sampler still leaves the target posterior distribution invariant. To see this, let $\boldsymbol{\theta}_i = (\theta_{1,i}, \theta_{2,i}, \theta_{3,i})'$ and $\boldsymbol{\theta}_{i+1} = (\theta_{1,i+1}, \theta_{2,i+1}, \theta_{3,i+1})'$ be two consecutive states. Then the construction of Algorithm 3(a) yields the following transition probability kernel:

$$T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) = \pi(\theta_{1,i+1}|\theta_{2,i}, D)\pi(\theta_{2,i+1}|\theta_{1,i+1}, D) \times \pi(\theta_{3,i+1}|\theta_{1,i+1}, \theta_{2,i+1}, D). \tag{2.5.1}$$

It follows that

$$\int_{R^3} T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1})\pi(\boldsymbol{\theta}_i|D)\, d\boldsymbol{\theta}_i = \pi(\boldsymbol{\theta}_{i+1}|D). \tag{2.5.2}$$

(The proof of (2.5.2) is left as an exercise.) Thus, $\pi(\boldsymbol{\theta}|D)$ is invariant with respect to the transition probability kernel $T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1})$. The modified version of the collapsed Gibbs sampler is useful and practically advantageous since drawing from the conditional posterior distributions is usually easier than sampling from the joint unconditional one. This is particularly true when dealing with higher-dimensional problems.

Using norms of the forward and backward operators of the induced Markov chain, Liu (1994) shows that the collapsed Gibbs works better than the grouped Gibbs, while the latter is better than the original Gibbs. It is expected that the collapsed Gibbs may work better than the modified collapsed Gibbs, while the modified version of collapsed Gibbs may be more beneficial than the original Gibbs. However, between the modified collapsed Gibbs and the grouped Gibbs, it is not straightforward to see which one works better. The performance of these two algorithms may depend on the correlations between θ_i and θ_j. If θ_1 and θ_2 are highly correlated, the grouped Gibbs is expected to work better. Otherwise, the modified collapsed Gibbs may have better performance.

The above three-component Gibbs sampler is also studied by Liu, Wong, and Kong (1994) and further discussed by Roberts and Sahu (1997), when the target distribution $\pi(\boldsymbol{\theta}|D)$ is normal. Regarding the grouping or blocking strategy for the Gibbs sampler, Roberts and Sahu (1997) provide a comprehensive study by comparing rates of convergence of various blocking combinations, and thus we refer the reader to their paper for further discussion. In general, grouping or blocking is beneficial, but often more computationally demanding. In particular, Roberts and Sahu (1997) show that if all partial correlations of a normal (Gaussian) target distribution are nonnegative, i.e., all of the off-diagonal elements of the inverse covariance matrix are nonpositive, then the grouped (blocked) Gibbs sampler has a faster rate of convergence than the standard (original) Gibbs sampler. That is, grouping positively correlated parameters in Gibbs sampling is always beneficial. However, Roberts and Sahu (1997) also find some ex-

amples showing that blocking can also make an algorithm converge more slowly.

2.5.2 *Reparameterizations: Hierarchical Centering and Rescaling*

As pointed out by Roberts and Sahu (1997), high correlations among the coordinates of θ diminish the speed of convergence of the Gibbs sampler (see also Hills and Smith 1992). The correlations among the coordinates are determined by the particular parameterization of the problem. Gelfand, Sahu, and Carlin (1995, 1996) argue that a hierarchically centered parameterization leads to faster mixing and convergence because it generally leads to smaller intercomponent correlations among the coordinates in Bayesian linear models. Roberts and Sahu (1997) further examine the hierarchically centered parameterization and they demonstrate that hierarchical centering yields faster mixing Gibbs samplers.

Here we illustrate this idea with a one-way analysis of variance model with random effects.

Example 2.5. One-way analysis of variance with random effects. Gelfand, Sahu, and Carlin (1996) and Roberts and Sahu (1997) consider the following one-way analysis of variance model. Assume that the error variance σ_e^2 is known and suppose that we have a single observation y_i for each population, i.e.,

$$y_i = \mu + \alpha_i + \epsilon_i, \quad i = 1, 2, \ldots, m, \tag{2.5.3}$$

where $\epsilon_i \sim N(0, \sigma_e^2)$, $\alpha_i \sim N(0, \sigma_\alpha^2)$, $\mu \sim N(\mu_0, \sigma_\mu^2)$, and σ_α^2 is also known. We denote the data by $D = \boldsymbol{y} = (y_1, y_2, \ldots, y_m)'$. Gelfand, Sahu, and Carlin (1996) rewrite (2.5.3) in a hierarchical form. Defining $\eta_i = \mu + \alpha_i$, we have

$$y_i|\eta_i \sim N(\eta_i, \sigma_e^2), \quad \eta_i|\mu \sim N(\mu, \sigma_\alpha^2), \quad \text{and} \quad \mu \sim N(\mu_0, \sigma_\mu^2).$$

This transformation from $(\alpha_1, \alpha_2, \ldots, \alpha_m)'$ to $(\eta_1, \eta_2, \ldots, \eta_m)'$ is thus referred to as hierarchical centering. Working in μ–η space, Gelfand, Sahu, and Carlin (1996) obtain

$$\text{corr}(\eta_i, \mu|D) = \left(1 + \frac{b\sigma_\alpha^2}{\sigma_e^2 \sigma_\mu^2}\right)^{-1/2} \tag{2.5.4}$$

and

$$\text{corr}(\eta_i, \eta_j|D) = \left(1 + \frac{b\sigma_\alpha^2}{\sigma_e^2 \sigma_\mu^2}\right)^{-1}, \tag{2.5.5}$$

where $b = \sigma_e^2 + \sigma_\alpha^2 + m\sigma_\mu^2$. The correlations given in (2.5.4) and (2.5.5) tend to 0 for fixed σ_e^2 and σ_μ^2 if $\sigma_\alpha^2 \to \infty$. On the other hand, if $\sigma_e^2 \to \infty$, the correlations do not approach 0, and in fact will tend to 1 if $\sigma_\mu^2 \to \infty$.

In μ–α space, we can obtain

$$\text{corr}(\alpha_i, \mu|D) = \left(1 + \frac{b\sigma_e^2}{\sigma_\alpha^2\sigma_\mu^2}\right)^{-1/2} \tag{2.5.6}$$

and

$$\text{corr}(\alpha_i, \alpha_j|D) = \left(1 + \frac{b\sigma_e^2}{\sigma_\alpha^2\sigma_\mu^2}\right)^{-1}. \tag{2.5.7}$$

The correlations given in (2.5.6) and (2.5.7) tend to 0 as $\sigma_e^2 \to \infty$, but do not approach 0 as $\sigma_\alpha^2 \to \infty$, and in fact will tend to 1 if $\sigma_\mu^2 \to \infty$ as well. In practice, when the random effects are needed, the error variance is much reduced. Thus σ_e^2 will rarely dominate the variability, so that the centered parameterization will likely be preferred. Roberts and Sahu (1997) obtain similar results by studying the rate of convergence of the Gibbs sampler.

Hierarchical centering is also useful for Bayesian nonlinear models. We will address this issue in Section 2.5.4 below. In the same spirit as hierarchical centering, hierarchical rescaling is another useful tool to reduce the correlations between the location coordinates and the scalar coordinates. We will illustrate hierarchical rescaling in the next subsection using ordinal response models.

2.5.3 Collapsing and Reparameterization for Ordinal Response Models

Consider a probit model for ordinal response data. Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)'$ denote an $n \times 1$ vector of n independent ordinal random variables. Assume that y_i takes a value of l $(1 \le l \le L,\ L > 2)$ with probability

$$p_{il} = \Phi(\gamma_l + \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{l-1} + \boldsymbol{x}_i'\boldsymbol{\beta}), \tag{2.5.8}$$

for $i = 1, ..., n$ and $l = 1, \ldots, L$, where $-\infty = \gamma_0 < \gamma_1 \le \gamma_2 < \ldots < \gamma_{L-1} < \gamma_L = \infty$, $\Phi(\cdot)$ denotes the $N(0,1)$ cumulative distribution function (cdf), which defines the link, $\boldsymbol{x}_i$ is a $p \times 1$ column vector of covariates, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is a $p \times 1$ column vector of regression coefficients. To ensure identifiability, we take $\gamma_1 = 0$. Let $\boldsymbol{\gamma} = (\gamma_2, \ldots, \gamma_{L-1})'$ and $D = (\boldsymbol{y}, X, n)$ denote the data, where X is the $n \times p$ design matrix with $\boldsymbol{x}_i'$ as its i^{th} row. Thus, the likelihood function is

$$L(\boldsymbol{\beta}, \boldsymbol{\gamma}|D) = \prod_{i=1}^{n} \left[\Phi(\gamma_{y_i} + \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{y_i-1} + \boldsymbol{x}_i'\boldsymbol{\beta})\right]. \tag{2.5.9}$$

We further assume that $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ has an improper uniform prior, i.e., $\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \propto 1$. The posterior distribution for $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ takes the form

$$\begin{aligned}\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}|D) &\propto L(\boldsymbol{\beta}, \boldsymbol{\gamma}|D)\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \\ &= \prod_{i=1}^{n} [\Phi(\gamma_{y_i} + \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{y_i-1} + \boldsymbol{x}_i'\boldsymbol{\beta})]\,. \end{aligned} \tag{2.5.10}$$

Chen and Shao (1999a) obtain necessary and sufficient conditions for the propriety of the posterior defined by (2.5.10). To facilitate the Gibbs sampler, Albert and Chib (1993) introduce latent variables z_i such that

$$y_i = l \text{ iff } \gamma_{l-1} \le z_i < \gamma_l,$$

for $l = 1, 2, \ldots, L$. Let $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)'$. The complete-data likelihood is given by

$$L(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{z}|D) \propto \prod_{i=1}^{n} [\exp\{-\tfrac{1}{2}(z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2\} 1\{\gamma_{y_i-1} \le z_i < \gamma_{y_i}\}], \tag{2.5.11}$$

where $1\{\gamma_{y_i-1} \le z_i < \gamma_{y_i}\}$ is the indicator function, and the joint posterior density for $(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{z})$ is given by

$$\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{z}|D) \propto \left\{ \prod_{i=1}^{n} [\exp\{-\tfrac{1}{2}(z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2\} 1\{\gamma_{y_i-1} \le z_i < \gamma_{y_i}\}] \right\}. \tag{2.5.12}$$

Then, Albert and Chib (1993) incorporate the unknown latent variables $\boldsymbol{z}$ as additional parameters to run the Gibbs sampler. The original Gibbs sampler for the ordinal probit model proposed by Albert and Chib (1993), which is referred to as the Albert–Chib algorithm thereafter, may be implemented as follows:

Albert–Chib Algorithm

Step 1. Draw $\boldsymbol{\beta}$ from

$$\boldsymbol{\beta}|\boldsymbol{z}, \boldsymbol{\gamma} \;\sim\; N((X'X)^{-1}X'\boldsymbol{z}, (X'X)^{-1}).$$

Step 2. Draw z_i from

$$z_i \;\sim\; N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1), \;\; \gamma_{y_i-1} \le z_i \le \gamma_{y_i}.$$

Step 3. Draw $\boldsymbol{\gamma}$ from

$$\gamma_l|\boldsymbol{\gamma}^{(-l)}, \boldsymbol{\beta}, \boldsymbol{z} \;\sim\; U[a_l, b_l], \tag{2.5.13}$$

where $a_l = \max\left\{\gamma_{l-1}, \max_{y_i=l} z_i\right\}$, $b_l = \min\left\{\gamma_{l+1}, \min_{y_i=l+1} z_i\right\}$, and $\boldsymbol{\gamma}^{(-l)}$ is $\boldsymbol{\gamma}$ with γ_l deleted.

Since, in Step 1 all p components of the regression coefficient vector are drawn simultaneously, the Albert–Chib algorithm is indeed a grouped Gibbs sampler. The implementation of the Albert–Chib algorithm is straightforward since the conditional posterior distributions are normal, truncated normal, or uniform. When the sample size n is not too big, the Albert–Chib algorithm works reasonably well. However, when n is large, say $n \geq 50$, slow convergence of the Albert–Chib algorithm may occur. Cowles (1996) points out this slow convergence problem. Because the interval (a_l, b_l) within which each γ_l must be generated from its full conditional can be very narrow, the cutpoint values may change very little between successive iterations, making the iterates highly correlated. Of course, slower convergence of the γ_l is also associated with the fact that the variance of the latent variables is fixed at one. The empirical study of Cowles (1996) further shows that the slow convergence of the cutpoints may also seriously affect the convergence of $\boldsymbol{\beta}$. To improve convergence of the original Gibbs sampler, she proposes a Metropolis–Hastings algorithm to generate the cutpoints from their conditional distributions; henceforth, this algorithm is called the Cowles algorithm. Instead of directly generating γ_l from (2.5.13), the Cowles algorithm generates $(\boldsymbol{\gamma}, \boldsymbol{z})$ jointly, which is essentially the same idea as the (modified) collapsed Gibbs sampler described in Section 2.5.1. The joint conditional distribution $\pi(\boldsymbol{\gamma}, \boldsymbol{z}|\boldsymbol{\beta}, D)$ can be expressed as the product of the marginal conditional distributions $\pi(\boldsymbol{\gamma}|\boldsymbol{\beta}, D)$ and $\pi(\boldsymbol{z}|\boldsymbol{\gamma}, \boldsymbol{\beta}, D)$. The Cowles algorithm can be described as follows:

Cowles Algorithm

Step 1. Draw $\boldsymbol{\beta}$ from

$$\boldsymbol{\beta}|\boldsymbol{z}, \boldsymbol{\gamma} \ \sim \ N((X'X)^{-1}X'\boldsymbol{z}, (X'X)^{-1}).$$

Step 2. Draw z_i from

$$z_i \ \sim \ N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1), \ \ \gamma_{y_i-1} \leq z_i \leq \gamma_{y_i}.$$

Step 3. Draw $\boldsymbol{\gamma}$ from the conditional distribution

$$\pi(\boldsymbol{\gamma}|\boldsymbol{\beta}, D) \propto \prod_{i=1}^{n} \left[\Phi(\gamma_{y_i} - \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{y_i-1} - \boldsymbol{x}_i'\boldsymbol{\beta})\right]. \qquad (2.5.14)$$

In the Cowles algorithm, a Metropolis–Hastings scheme is used to draw $\boldsymbol{\gamma}$. That is, given the value $\boldsymbol{\gamma}_{j-1}$ from the previous iteration, a vector of proposal cutpoint values, γ_l^*, $l = 2, 3, \ldots, L-1$, is generated from a truncated normal distribution

$$\gamma_l^*|\gamma_{l-1}^*, \gamma_{l+1,j-1} \ \sim \ N(\gamma_{l,j-1}, \sigma_\gamma^2), \qquad (2.5.15)$$

where $\gamma_{l-1}^* \leq \gamma_l^* \leq \gamma_{l+1,j-1}$. The acceptance probability for the vector $\boldsymbol{\gamma}^*$ of new cutpoints is $a = \min\{1, R\}$, where

$$R = \prod_{l=2}^{L-1} \frac{\{\Phi\{(\gamma_{l+1,j-1} - \gamma_{l,j-1})/\sigma_\gamma\} - \Phi\{(\gamma_{l-1}^* - \gamma_{l,j-1})/\sigma_\gamma\}}{\Phi\{(\gamma_{l+1}^* - \gamma_l^*)/\sigma_\gamma\} - \Phi\{(\gamma_{l-1,j-1} - \gamma_l^*)/\sigma_\gamma\}}$$
$$\times \prod_{i=1}^{n} \frac{\Phi(\gamma_{y_i}^* - \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{y_i-1}^* - \boldsymbol{x}_i'\boldsymbol{\beta})}{\Phi(\gamma_{y_i,j-1} - \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{y_i-1,j-1} - \boldsymbol{x}_i'\boldsymbol{\beta})}. \tag{2.5.16}$$

However, our experience suggests that in the Cowles algorithm, the truncated normal-distribution in (2.5.15) might not serve as a good proposal density for the conditional posterior density in (2.5.14), since it is not spread out enough (see Tierney (1994), and Section 2.2). Further, Cowles (1996) points out that a good σ_γ^2 in (2.5.15) is difficult to obtain even when using a conventional updating scheme.

To overcome the difficulties arising in the Cowles algorithm, Nandram and Chen (1996) develop an improved algorithm using a Dirichlet proposal distribution based on a rescaling transformation. Similar to hierarchical centering, the hierarchically rescaled transformations proposed by Nandram and Chen (1996) are

$$\delta = 1/\gamma_{L-1}, \;\; \boldsymbol{\gamma}^* = \delta\boldsymbol{\gamma}, \;\; \boldsymbol{\beta}^* = \delta\boldsymbol{\beta}, \;\; \text{and} \;\; \boldsymbol{z}^* = \delta\boldsymbol{z}. \tag{2.5.17}$$

Note that in (2.5.17), $\gamma_0^* = -\infty < \gamma_1^* = 0 \leq \gamma_2^* \leq \cdots \leq \gamma_{L-2}^* \leq \gamma_{L-1}^* = 1 < \gamma_L^* = \infty$, and that effectively there are only $L-3$ unknown cutpoints in the reparameterized model. Let $\boldsymbol{\gamma}^* = (\gamma_2^*, \gamma_3^*, \ldots, \gamma_{L-2}^*)'$. Thus, when $L = 3$, there are no unknown cutpoints in $\boldsymbol{\gamma}^*$, which is advantageous when one deals with a 3-level ordinal response model. For $L = 3$, the Nandram–Chen algorithm can be implemented as follows:

Nandram–Chen Algorithm

Step 1. Draw $\boldsymbol{\beta}^*$ from

$$\boldsymbol{\beta}^* | \boldsymbol{z}^*, \boldsymbol{\gamma}^* \;\sim\; N((X'X)^{-1}X'\boldsymbol{z}, \delta^2(X'X)^{-1}).$$

Step 2. Draw z_i^* from

$$z_i^* | \boldsymbol{\beta}^*, \delta \;\sim\; N(\boldsymbol{x}_i'\boldsymbol{\beta}^*, \delta^2), \;\; \gamma_{y_i-1}^* \leq z_i^* < \gamma_{y_i}^*.$$

Step 3. Draw δ^2 from

$$\delta^2 | \boldsymbol{\beta}^*, \boldsymbol{z}^* \;\sim\; \mathcal{IG}\left\{\frac{n+p+L-2}{2}, \tfrac{1}{2}[(\boldsymbol{z}^* - X\boldsymbol{\beta}^*)'(\boldsymbol{z}^* - X\boldsymbol{\beta}^*)]\right\}.$$

For $L > 3$, the Nandram–Chen algorithm requires an additional step to draw $\boldsymbol{\gamma}^*$. That is,

Step 4. Draw $\boldsymbol{\gamma}^*$ from the conditional posterior distribution

$$\pi(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^*, \delta^2, D)$$
$$\propto \prod_{i=1}^{n} \left\{ \Phi\left(\frac{\gamma^*_{y_i} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right) - \Phi\left(\frac{\gamma^*_{y_i-1} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right) \right\}. \tag{2.5.18}$$

Instead of using a truncated normal proposal density as in the Cowles algorithm, Nandram and Chen (1996) construct a Dirichlet proposal density for $\pi(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^*, \delta^2, D)$. The motivation for the Dirichlet proposal density is given as follows. Let $q_{l-1} = \gamma^*_l - \gamma^*_{l-1}$, $l = 2, \ldots, L-1$, and let $\boldsymbol{q} = (q_1, q_2, \ldots, q_{L-2})'$, $q_l \geq 0$, $l = 1, 2, \ldots, L-2$ and $\sum_{l=1}^{L-2} q_l = 1$. By the fundamental mean value theorem,

$$\Phi\left(\frac{\gamma^*_{y_i} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right) - \Phi\left(\frac{\gamma^*_{y_i-1} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right) = \frac{1}{\delta}\phi\left(\frac{\xi_{y_i} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right) q_{y_i-1}, \tag{2.5.19}$$

where ξ_{y_i} is a real number between $\gamma^*_{y_i}$ and $\gamma^*_{y_i-1}$, $i = 1, 2, 3, \ldots, n$, and $\phi(\cdot)$ is the standard normal density function. Then by (2.5.19),

$$\pi(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^*, \delta^2, D) \propto g_1(\xi) g_2(\boldsymbol{q}), \tag{2.5.20}$$

where

$$g_1(\xi) = \prod_{i=1}^{n} \phi\left(\frac{\xi_{y_i} - \boldsymbol{x}_i'\boldsymbol{\beta}^*}{\delta}\right), \quad g_2(\boldsymbol{q}) = \prod_{l=1}^{L-2} q_l^{n_{l+1}}, \quad \text{and} \quad n_l = \sum_{i=1}^{n} 1\{y_i = l\}.$$

for $l = 1, 2, \ldots, L$. While in the Cowles algorithm the proposal density is based on $g_1(\xi)$, Nandram and Chen (1996) use $g_2(\boldsymbol{q})$ to construct a proposal density. This is quite natural because if there are no covariates, we can associate $\boldsymbol{q}$ with the bin "probabilities." Assigning an improper prior to the bins and treating the bin counts as data, the joint posterior distribution of these bin "probabilities" is a Dirichlet distribution with the bin counts as the posterior parameters.

An approximation of $\pi(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^*, \delta^2, D)$ motivated by (2.5.20) is

$$\pi(\boldsymbol{q}|\boldsymbol{\beta}^*, \delta^2, D) \propto \prod_{l=1}^{L-2} q_l^{\alpha_l n_{l+1} - 1}, \tag{2.5.21}$$

where $0 \leq \alpha_l \leq 1$, $l = 1, \ldots, L-2$, are the tuning parameters. (That is, $\boldsymbol{q}$ has a Dirichlet distribution.) The proposal density (2.5.21) is attractive because we can draw the entire vector $\boldsymbol{q}$ at once, and it does not depend on $\boldsymbol{\beta}^*$ and δ^2. In addition, the Dirichlet proposal density will be more useful when more complex link functions (e.g., logistic and complementary log-log) are used. Moreover, one can choose the α_l in (2.5.21) by taking the α_l so as to make the dispersion of the posterior distribution of $\boldsymbol{q}$ comparable to or at least as large as that of the distribution of $\boldsymbol{\gamma}^*$. The rest of the implementation for drawing $\boldsymbol{\gamma}^*$ simultaneously from its conditional posterior

distribution is the same as the one given in the Cowles algorithm, and thus the details are omitted.

Nandram and Chen (1996) conduct several simulation studies, and their empirical results show that the Nandram–Chen algorithm substantially improves convergence of the Gibbs sampler compared to the Albert–Chib and Cowles algorithms. A partial explanation for this is that:

(a) hierarchical rescaling reduces the correlations between the cutpoints and the latent variables; and

(b) the meaningful choice (from a theoretical or statistical viewpoint) of the proposal density has better properties than the truncated normal proposal density used in the Cowles algorithm.

The Nandram–Chen algorithm works well if the cell counts n_l are relatively balanced. When the cell counts are unbalanced, in particular, if some of those counts are close to zero, $\pi(\boldsymbol{q}|\boldsymbol{\beta}^*, \delta^2, D)$ in (2.5.21) may not serve as a good proposal density. For these cases, Chen and Dey (1996) propose a Metropolis–Hastings algorithm using a "de-constraint" transformation to draw $\boldsymbol{\gamma}^*$. A similar transformation of the cutpoints is also considered in Albert and Chib (1998). Let

$$\gamma_l^* = \frac{\gamma_{l-1}^* + e^{\zeta_l}}{1 + e^{\zeta_l}}, \quad l = 2, \ldots, L-2, \tag{2.5.22}$$

and $\boldsymbol{\zeta} = (\zeta_2, \ldots, \zeta_{L-2})'$. Then the conditional posterior distribution for ζ is

$$\pi(\boldsymbol{\zeta}|\boldsymbol{\beta}^*, \delta^2, D) \propto \pi(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^*, \delta^2, D) \prod_{l=2}^{L-2} \frac{(1-\gamma_{l-1})e^{\zeta_l}}{(1+e^{\zeta_l})^2}, \tag{2.5.23}$$

where $\boldsymbol{\gamma}^*$ is evaluated at $\gamma_l^* = (\gamma_{l-1}^* + e^{\zeta_l})/(1+e^{\zeta_l})$ for $l = 2, 3, \ldots, L-2$. The remaining steps of the Metropolis–Hastings algorithm are the same as the algorithm for sampling ρ as described in Example 2.3. This modified Nandram–Chen algorithm is thus called the *Chen–Dey algorithm.*

2.5.4 Hierarchical Centering for Poisson Random Effects Models

A Poisson regression model with AR(1) random effects is used for modeling the pollen count data in Example 1.3. Using (1.3.3), the complete-data likelihood is given by

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D) = &\exp\{\boldsymbol{y}'(X\boldsymbol{\beta} + \boldsymbol{\epsilon}) - J_n' Q(\boldsymbol{\beta}, \boldsymbol{\epsilon}) - J_n' C(\boldsymbol{y})\} \\ &\times (2\pi\sigma^2)^{-n/2}(1-\rho^2)^{-(n-1)/2} \\ &\times \exp\left\{-\frac{1}{2\sigma^2}\boldsymbol{\epsilon}'\Sigma^{-1}\boldsymbol{\epsilon}\right\}, \end{aligned} \tag{2.5.24}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)'$, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)'$, X is the $n \times 8$ matrix of covariates with the t^{th} row equal to $\boldsymbol{x}_t'$, J_n is an $n \times 1$ vector of ones, and $Q(\boldsymbol{\beta}, \boldsymbol{\epsilon})$ is an $n \times 1$ vector with the t^{th} element equal to $q_t = \exp\{\epsilon_t + \boldsymbol{x}_t'\boldsymbol{\beta}\}$, $C(\boldsymbol{y})$ is an $n \times 1$ vector with the j^{th} element $\log(y_j!)$, and $D = (n, \boldsymbol{y}, X)$. In (2.5.24), $\Sigma = (\sigma_{ij})$ with $\sigma_{ij} = \rho^{|i-j|}$, where $\rho^{|i-j|}$ is the correlation between (ϵ_i, ϵ_j), and $-1 \leq \rho \leq 1$. Assume that a noninformative prior for $(\boldsymbol{\beta}, \sigma^2, \rho)$ has the form

$$\pi(\boldsymbol{\beta}, \sigma^2, \rho) \propto (\sigma^2)^{-(\delta_0+1)} \exp(-\sigma^{-2}\gamma_0), \tag{2.5.25}$$

where the hyperparameters $\delta_0 > 0$ and $\gamma_0 > 0$ are prespecified. Then, the joint posterior distribution for $(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon})$ is given by

$$\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D) \propto L(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D)(\sigma^2)^{-(\delta_0+1)} \exp(-\sigma^{-2}\gamma_0), \tag{2.5.26}$$

where the likelihood $L(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D)$ is defined by (2.5.24). It can be shown that if X^* is of full rank, where X^* is a matrix induced by X and $\boldsymbol{y}$ with its t^{th} row equal to $1\{y_t > 0\}\boldsymbol{x}_t'$, then the posterior distribution $\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D)$ is proper.

Unlike the one-way analysis of variance model with random effects in Example 2.5, the Poisson regression model with AR(1) random effects is not a linear model. The exact correlations among the parameters $\boldsymbol{\epsilon}$, $\boldsymbol{\beta}$, σ^2, and ρ are not clear. However, it is expected that the correlation patterns in the Poisson regression model are similar to that of the one-way analysis of variance model. Ibrahim, Chen, and Ryan (1999) observe that the original Gibbs sampler without hierarchical centering results in very slow convergence and poor mixing. In particular, the correlation ρ appears to converge the slowest. They further find that the hierarchical centering technique is perfectly suited for this problem, and appears quite crucial for convergence of the Gibbs sampler.

Similar to the one-way analysis of variance model, a hierarchically centered reparameterization is given by

$$\boldsymbol{\eta} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2.5.27}$$

Using (2.5.26), the reparameterized posterior for $(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\eta})$ is written as

$$\begin{aligned}\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\eta}|D) \propto & \exp\{\boldsymbol{y}'\boldsymbol{\eta} - J_n'Q(\boldsymbol{\eta}) - J_n'C(\boldsymbol{y})\} \\ & \times (2\pi\sigma^2)^{-n/2}(1-\rho^2)^{-(n-1)/2} \\ & \times \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\eta} - X\boldsymbol{\beta})'\Sigma^{-1}(\boldsymbol{\eta} - X\boldsymbol{\beta})\right\},\end{aligned} \tag{2.5.28}$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)'$, and $Q(\boldsymbol{\eta})$ is an $n \times 1$ vector with the t^{th} element equal to $q_t = \exp(\eta_t)$.

The Gibbs sampler for sampling from the reparameterized posterior $\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\eta}|D)$ requires the following steps:

Step 1. Draw $\boldsymbol{\eta}$ from its conditional posterior distribution

$$\pi(\boldsymbol{\eta}|\boldsymbol{\beta},\sigma^2,\rho,D) \propto \exp\left\{\boldsymbol{y}'\boldsymbol{\eta} - J_n' Q(\boldsymbol{\eta}) - \frac{(\boldsymbol{\eta}-X\boldsymbol{\beta})'\Sigma^{-1}(\boldsymbol{\eta}-X\boldsymbol{\beta})}{2\sigma^2}\right\}. \tag{2.5.29}$$

Step 2. Draw $\boldsymbol{\beta}$ from

$$\boldsymbol{\beta}|\boldsymbol{\eta},\sigma^2,\rho,D \;\sim\; N_8((X'\Sigma^{-1}X)^{-1}X'\Sigma^{-1}\boldsymbol{\eta},\sigma^2(X'\Sigma^{-1}X)^{-1}).$$

Step 3. Draw σ^2 from its conditional posterior

$$\sigma^2|\boldsymbol{\beta},\rho,\boldsymbol{\eta},D \;\sim\; \mathcal{IG}(\delta^*,\gamma^*),$$

where $\delta^* = \delta_0 + n/2$, $\gamma^* = \gamma_0 + \frac{1}{2}(\boldsymbol{\eta}-X\boldsymbol{\beta})'\Sigma^{-1}(\boldsymbol{\eta}-X\boldsymbol{\beta})$, and $\mathcal{IG}(\delta^*,\gamma^*)$ is an inverse gamma distribution.

Step 4. Draw ρ from its conditional posterior

$$\pi(\rho|\sigma^2,\boldsymbol{\beta},\boldsymbol{\eta},D)$$
$$\propto (1-\rho^2)^{-(n-1)/2}\exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\eta}-X\boldsymbol{\beta})'\Sigma^{-1}(\boldsymbol{\eta}-X\boldsymbol{\beta})\right\}.$$

In Step 1, it can be shown that $\pi(\boldsymbol{\eta}|\boldsymbol{\beta},\sigma^2,\rho,D)$ is log-concave in each component of $\boldsymbol{\eta}$ (see Exercise 2.7). Thus $\boldsymbol{\eta}$ can be drawn using the adaptive rejection sampling algorithm of Gilks and Wild (1992). The implementation of Steps 2 and 3 is straightforward, which may be a bonus of hierarchical centering, since sampling $\boldsymbol{\beta}$ is much more expensive before the reparameterization. In Chapter 9, we will also show that the hierarchical centering reparameterization can greatly ease the computational burden in Bayesian variable selection. In Step 4, we can use the algorithm in Example 2.3 for sampling ρ.

2.6 Acceleration Algorithms for MCMC Sampling

The major problems for many MCMC algorithms are slow convergence and poor mixing. For example, the Gibbs sampler converges slowly even for a simple ordinal response model as discussed in Section 2.5.3. In the earlier sections of this chapter, we discuss several tools for speeding up an MCMC algorithm, which include grouping (blocking) and collapsing (Liu 1994; Liu, Wong, and Kong 1994; Roberts and Sahu 1997), reparameterizations (Gelfand, Sahu, and Carlin 1995 and 1996; Roberts and Sahu 1997). The other useful techniques are adaptive direction sampling (Gilks, Roberts, and George 1994; Roberts and Gilks 1994), Multiple-Try Metropolis (Liu, Liang, and Wong 1998a), auxiliary variable methods (Besag and Green 1993; Damien, Wakefield, and Walker 1999), simulated tempering (Marinari and Parisi 1992; Geyer and Thompson 1995), and working parameter methods (Meng and van Dyk 1999). In this section, we present

two special acceleration MCMC algorithms, i.e., grouped move and Multigrid MC sampling (Liu and Wu 1997; Liu and Sabatti 1998 and 1999) and covariance-adjusted MCMC sampling (Liu 1998), which provide us with a general framework of how to further improve mixing and convergence of an MCMC algorithm.

2.6.1 Grouped Move and Multigrid Monte Carlo Sampling

Goodman and Sokal (1989) present a comparative review of the multigrid Monte Carlo (MGMC) method, which is a stochastic generalization of the multigrid (MG) method for solving finite-difference equations. Liu and Wu (1997) and Liu and Sabatti (1998 and 1999) generalize Goodman and Sokal's MGMC via groups of transformations with applications to MCMC sampling. They propose a Grouped Move Multigrid Monte Carlo (GM-MGMC) algorithm and a generalized version of the MGMC algorithm for sampling from a target posterior distribution.

Assume that the target posterior distribution $\pi(\boldsymbol{\theta}|D)$ is defined on Ω and that an MCMC algorithm such as the Gibbs sampler or Metropolis–Hastings algorithm is used to generate a Markov chain $\{\boldsymbol{\theta}_i,\ i = 0, 1, 2, \dots\}$ from the target distribution $\pi(\boldsymbol{\theta}|D)$. We call the MCMC algorithm used to generate the $\boldsymbol{\theta}_i$ the *parent* MCMC algorithm. Let Γ be a locally compact transformation group (Rao 1987) on Ω. Then the GM-MGMC algorithm of Liu and Wu (1997) and Liu and Sabatti (1998) proceeds as follows:

GM-MGMC Algorithm

MCMC Step. Generate an iteration $\boldsymbol{\theta}_i$ from the parent MCMC.

GM Step. Draw the group element g from

$$g \ \sim \ \pi(g|\boldsymbol{\theta}_i)H(g) \propto \pi(g(\boldsymbol{\theta}_i)|D)J_g(\boldsymbol{\theta}_i)H(dg), \tag{2.6.1}$$

and *adjust*

$$\boldsymbol{\theta}_i \leftarrow g(\boldsymbol{\theta}_i).$$

In (2.6.1) $H(dg)$ is the right-invariant Haar measure on Ω and $J_g(\boldsymbol{\theta}_i)$ is the Jacobian of g evaluated at $\boldsymbol{\theta}_i$. Liu and Wu (1997) show that if Γ is a locally compact group of transformations for $\boldsymbol{\theta} \in \Omega$ with a unimodular right-invariant Haar measure $H(dg)$, then $g(\boldsymbol{\theta}_i) \sim \pi(\boldsymbol{\theta}|D)$, provided $\boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta}|D)$ and $g \sim \pi(g|\boldsymbol{\theta}_i)$. This result ensures that the target posterior distribution $\pi(\boldsymbol{\theta}|D)$ is the stationary distribution of the Markov chain induced by the GM-MGMC algorithm. As discussed in Liu and Sabatti (1998), the GM step is flexible: by selecting appropriate groups of transformations one can achieve either the effect of reparameterization or that of "blocking" or "grouping." Liu and Sabatti use several examples to illustrate these points.

In many cases, directly sampling g in the GM step may be difficult or expensive to achieve. For these cases, Liu and Sabatti (1998, 1999) propose a Markov transition. Assume that $T_{\boldsymbol{\theta}}(g', g)H(dg)$ is a Markov transition function, which leaves (2.6.1) invariant and satisfies the *transformation-invariance*, i.e.,

$$T_{\boldsymbol{\theta}}(g', g) = T_{g_0^{-1}\boldsymbol{\theta}}(g'g_0, gg_0) \tag{2.6.2}$$

for all g, g', and g_0 in Γ. Then, the GM-MGMC algorithm can be extended to the following Generalized MGMC algorithm:

Generalized MGMC Algorithm

MCMC Step. Generate an iteration $\boldsymbol{\theta}_i$ from the parent MCMC.

GM Step. Draw the group element g from

$$g \sim T_{\boldsymbol{\theta}_i}(I, g), \tag{2.6.3}$$

and *adjust*

$$\boldsymbol{\theta}_i \leftarrow g(\boldsymbol{\theta}_i).$$

In (2.6.3), I denotes the identity of the group. Similar to the GM-MGMC algorithm, it can be shown that $g(\boldsymbol{\theta}) \sim \pi(\boldsymbol{\theta}|D)$ provided $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}|D)$.

In the GM-MGMC algorithm or the generalized MGMC algorithm, one GM step is used. In some situations, multiple GM steps can also be applied. For example, a three-step GM algorithm can be described by the following cycle. Starting with $\boldsymbol{\theta}_i \in \Omega$, which is drawn from a parent MCMC algorithm:

(i) draw g from a proper $T_{\boldsymbol{\theta}_i}(I, g)$, which leaves (2.6.1) invariant and satisfies (2.6.2), and update $\boldsymbol{\theta}^* = g(\boldsymbol{\theta})$;

(ii) draw g^* from a proper $T_{\boldsymbol{\theta}^*}(I, g^*)$ and update $\boldsymbol{\theta}^{**} = g^*(\boldsymbol{\theta}^*)$; and

(iii) draw g^{**} from $T_{\boldsymbol{\theta}^{**}}(I, g^{**})$ and update $\boldsymbol{\theta}_{i+1} = g^{**}(\boldsymbol{\theta}^{**})$.

Then, if $\boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta}|D)$, then $\boldsymbol{\theta}_{i+1} \sim \pi(\boldsymbol{\theta}|D)$.

The GM-MCMC algorithm is a flexible generalization of the Gibbs sampler or the Metropolis–Hastings algorithm, which enables us to design more efficient MCMC algorithms. The purpose of the GM step is to improve the convergence or mixing rates of the parent MCMC algorithm. The nature of the multiple GM steps allows us to achieve such an improvement in an iterative fashion. That is, if the performance of a parent MCMC algorithm is unsatisfactory, one can design a GM step, and then make an additional draw in each iteration of the parent algorithm. From an implementational standpoint, the GM step requires only adding a subroutine to the existing code and does not require a change in the basic structure of the code. After a one-step adjustment, the new MCMC algorithm induced by the GM step

can be viewed as a new parent algorithm. Then, a similar adjustment can be applied to this new parent algorithm. One can repeat this procedure many times until a satisfactory convergence rate is achieved. Therefore, accelerating an MCMC algorithm can be viewed as a continuous improvement process.

Although the GM-MCMC algorithm provides a general framework for speeding up an MCMC algorithm, finding a computationally feasible group of transformations along with a unimodular right-invariant Haar measure $H(dg)$ is a difficult task. Two simple groups are the multiplicative group, i.e., $g(\boldsymbol{\theta}) = g\boldsymbol{\theta}$, and the additive group, i.e., $g(\boldsymbol{\theta}) = \boldsymbol{\theta} + \boldsymbol{g}$. For these two special cases, the associated unimodular right-invariant Haar measures are $H(dg) = 1/g$ for the multiplicative group and $H(d\boldsymbol{g}) = 1$ for the additive group. Although both the multiplicative group and additive group result in unimodular Haar measures, the linear combination of these two group transformations, i.e., $g(\boldsymbol{\theta}) = g_1\boldsymbol{\theta} + \boldsymbol{g}_2$, does not yield a unimodular Haar measure (see Nachbin 1965). In addition, GM-MGMC may not always improve convergence over the parent MCMC algorithm. In fact, Liu and Sabatti (1998) provide an example showing that GM-MGMC can result in a slower convergence rate than the parent MCMC algorithm. However, in many cases, GM-MGMC can achieve a substantial improvement in the convergence and mixing rate over a parent MCMC algorithm; see Liu and Sabatti (1998) and Chen and Liu (1999) for several illustrative examples. In practice, GM-MGMC can be viewed as an advanced experimental technique for improving convergence of an MCMC algorithm, and it can be helpful in getting a better understanding of the problem. As a practical guideline, we suggest using a GM step as long as it is simple and easy to implement.

2.6.2 Covariance-Adjusted MCMC Algorithm

Liu (1998) provides an alternative method for speeding up an MCMC algorithm using the idea of covariance adjustment. Let $\{\boldsymbol{\theta}_i,\ i = 0, 1, 2, \ldots\}$ be generated by the parent MCMC algorithm, having the stationary distribution $\pi(\boldsymbol{\theta}|D)$. Also let $(\boldsymbol{\xi}, \boldsymbol{\delta}) = \mathcal{M}(\boldsymbol{\theta})$ be a one-to-one mapping from Ω on which the target distribution is defined onto the space $\Xi \times \Delta$. Then, the covariance-adjusted MCMC (CA-MCMC) algorithm at the i^{th} iteration consists of the following two steps:

CA-MCMC Algorithm

MCMC Step. Generate an iteration $\boldsymbol{\theta}_i$ from the parent MCMC and compute $(\boldsymbol{\xi}_i, \boldsymbol{\delta}_i) = \mathcal{M}(\boldsymbol{\theta}_i)$.

CA Step. Draw $\boldsymbol{\delta}_i^*$ from the conditional posterior distribution $\pi(\boldsymbol{\delta}|\boldsymbol{\xi}_i, D)$ and *adjust* $\boldsymbol{\theta}_i$ by

$$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i^* = \mathcal{M}^{-1}(\boldsymbol{\xi}_i, \boldsymbol{\delta}_i^*), \tag{2.6.4}$$

where $\mathcal{M}^{-1}(\boldsymbol{\xi}, \boldsymbol{\delta})$ is the inverse mapping of $(\boldsymbol{\xi}, \boldsymbol{\delta}) = \mathcal{M}(\boldsymbol{\theta})$.

Liu (1998) shows that the CA-MCMC algorithm converges to the target distribution $\pi(\boldsymbol{\theta}|D)$. That is, if the Markov chain induced by an MCMC algorithm is irreducible, aperiodic, and stationary with the equilibrium distribution $\pi(\boldsymbol{\theta}|D)$, so is the covariance-adjusted Markov chain. We refer the reader to Liu's paper for a formal proof. This result ensures that the CA step guarantees the correctness of the CA-MCMC algorithm. In addition, Liu (1998) also proves that the CA-MCMC algorithm converges at least as fast as its parent MCMC algorithm in the sense that the CA-MCMC algorithm results in a smaller reversed Kullback–Leibler information distance (e.g., Liu, Wong, and Kong 1995). This implies that the Markov sequence induced by the CA-MCMC algorithm has less dependence than that induced by the parent MCMC algorithm. This result essentially distinguishes the CA-MCMC algorithm from the GM-MGMC algorithm since the latter does not always guarantee faster convergence than its parent MCMC algorithm.

The key issue in using the CA-MCMC algorithm is how to construct the $\boldsymbol{\delta}$-variable so that the resulting algorithm is efficient and simple to implement. A general strategy proposed by Liu (1998) is to construct the $\boldsymbol{\delta}$-variable based on parameters and their sufficient statistics. We use an example given in Liu (1998) to illustrate this idea.

Example 2.6. One-way analysis of variance with random effects (Example 2.5 continued). Consider the one-way analysis of the variance model given in Example 2.5. Assume that the error variance σ_e^2 is known and that a single observation y_i for each population, i.e.,

$$y_i = \mu + \alpha_i + \epsilon_i, \quad i = 1, 2, \ldots, m, \tag{2.6.5}$$

where $\epsilon_i \sim N(0, \sigma_e^2)$, $\alpha_i \sim N(0, \sigma_\alpha^2)$, and σ_α^2 is also known. We assume that $\pi(\mu) \propto 1$ and let $\bar{y} = (1/m) \sum_{i=1}^m y_i$ and $D = (y_1, y_2, \ldots, y_m)$.

For this one-way analysis of the variance model, the vector of model parameters is $\boldsymbol{\theta} = (\mu, \alpha_1, \alpha_2, \ldots, \alpha_m)'$. We use the Gibbs sampler as the parent MCMC algorithm. To apply the CA-MCMC algorithm, we need to construct $\boldsymbol{\xi}$ and $\boldsymbol{\delta}$. In this example, μ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)'$ may be highly correlated (see (2.5.6) and (2.5.7)), which may cause slow convergence of the original Gibbs sampler. To break down this correlation pattern, we consider the parameter μ. From (2.6.5), it is easy to see that,

a posteriori, the sufficient statistic for μ is

$$\bar{\alpha} = \frac{1}{m}\sum_{i=1}^{m} \alpha_i.$$

Let $\xi_i = \alpha_i - \bar{\alpha}$. Define

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_m)' \text{ and } \boldsymbol{\delta} = (\mu, \bar{\alpha})', \tag{2.6.6}$$

and let $\Xi = \{\xi : \sum_{i=1}^{m} \xi_i = 0,\ \xi_i \in R \text{ for } i = 1, 2, \ldots, m\}$. Then, (2.6.6) clearly defines a one-to-one mapping from R^{m+1} to $\Xi \times R^2$. The Jacobian of this transformation is $J_{(\mu,\alpha)\to(\mu,\bar{\alpha},\xi_1,\ldots,\xi_{m-1})} = 1$. The CA step requires drawing a $(\mu, \bar{\alpha})$ conditional on ξ_i for $i = 1, 2, \ldots, m$. The complete CA-MCMC algorithm can be stated as follows:

CA-MCMC for One-Way Analysis of Variance with Random Effects

Gibbs Step. Draw $(\mu|\boldsymbol{\alpha}, D) \sim N(\bar{y} - \bar{\alpha}, \sigma_e^2/m)$ and

$$(\alpha_i|\mu, D) \sim N\left(\frac{\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}(y_i - \mu), \frac{\sigma_e^2\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}\right).$$

CA Step. Draw $(\bar{\alpha}^*|\boldsymbol{\xi}, D) \sim N(0, \sigma_\alpha^2/m)$ and

$$(\mu^*|\bar{\alpha}^*, \boldsymbol{\xi}, D) \sim N\left(\bar{y} - \bar{\alpha}^*, \frac{\sigma_e^2}{m}\right),$$

then *adjust*

$$\mu \leftarrow \mu^* \text{ and } \alpha_i \leftarrow \xi_i + \bar{\alpha}^* \text{ for } i = 1, 2, \ldots, m.$$

From the structure of the above CA-MCMC algorithm, it can be seen that the draws of $(\mu^*, \bar{\alpha}^*, \xi_1 + \bar{\alpha}^*, \xi_2 + \bar{\alpha}^*, \ldots, \xi_m + \bar{\alpha}^*)$ are independent. Thus, the rate of convergence of this CA-MCMC algorithm is 0. Roberts and Sahu (1997) show that the rate of convergence of the Markov chain using the Gibbs step only is $\sigma_\alpha^2/(\sigma_\alpha^2 + \sigma_e^2)$ and that the rate of convergence of the Gibbs sampler based on the hierarchically centered transformation given in Example 2.5 (namely, $\mu = \mu$ and $\eta_i = \mu + \alpha_i$ for $i = 1, 2, \ldots, m$) is $\sigma_e^2/(\sigma_\alpha^2 + \sigma_e^2)$. Thus, CA-MCMC sampling outperforms original Gibbs sampling as well as hierarchical centering. This simple example also illustrates another important feature of the CA-MCMC algorithm, specifically, the concept of sufficient statistics, which can be nicely integrated into MCMC sampling and dramatically improves convergence of the MCMC algorithm.

2.6.3 An Illustration

To illustrate how an MCMC algorithm can be adjusted to achieve faster convergence and better mixing, we consider the following ordinal response data problem. The data are given in Table 2.1.

TABLE 2.1. The Rating Data.

Gender	F	M	F	M	F	M	F	M
Rating	good	fair	good	poor	good	poor	good	good

We code female as $X = 1$ and male as $X = 0$ and we also denote the response (Y) to be 1 for "poor," 2 for "fair," and 3 for "good." We let $\boldsymbol{y} = (y_1, y_2, \ldots, y_8)'$ denote a 8×1 vector of n independent ordinal responses. Assume that y_i takes a value of l $(1 \leq l \leq 3)$ with probability

$$p_{il} = \Phi(\gamma_l + \boldsymbol{x}_i'\boldsymbol{\beta}) - \Phi(\gamma_{l-1} + \boldsymbol{x}_i'\boldsymbol{\beta}),$$

for $i = 1, ..., 8$ and $l = 1, 2, 3$, where $-\infty = \gamma_0 < \gamma_1 \leq \gamma_2 < \gamma_3 = \infty$, x_i is a 2×1 column vector of covariates denoting the intercept and gender, and $\boldsymbol{\beta} = (\beta_0, \beta_1)'$ is a 2×1 column vector of regression coefficients. To ensure identifiability, we fix $\gamma_1 = 0$. Let $D = (\boldsymbol{y}, X)$, where X is the 8×2 design matrix with $\boldsymbol{x}_i'$ as its i^{th} row.

Using (2.5.12), the complete-data likelihood is

$$L(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z}|D) \propto \prod_{i=1}^{8} [\exp\{-\tfrac{1}{2}(z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2\} 1\{\gamma_{y_i - 1} \leq z_i < \gamma_{y_i}\}], \qquad (2.6.7)$$

where $\boldsymbol{z} = (z_1, z_2, \ldots, z_8)'$ is the vector of latent variables such that $y_i = l$ if $\gamma_{l-1} \leq z_i < \gamma_l$ for $l = 1, 2, 3$ and $i = 1, 2, \ldots, 8$. Consider a prior distribution for $(\boldsymbol{\beta}, \gamma_2)$ taking the form

$$\pi(\boldsymbol{\beta}, \gamma_2) \propto \exp\Big\{ -\frac{\tau}{2}\boldsymbol{\beta}'\boldsymbol{\beta} \Big\}, \qquad (2.6.8)$$

where $\tau > 0$ is a known precision parameter. Here we take $\tau = 0.001$. Using (2.6.7) and (2.6.8), the posterior for $(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z})$ is given by

$$\pi(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z}|D) \propto L(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z}|D)\pi(\boldsymbol{\beta}, \gamma_2). \qquad (2.6.9)$$

Using the necessary and sufficient conditions of Chen and Shao (1999a), it can be shown that when $\pi(\boldsymbol{\beta}, \gamma_2) \propto 1$, the posterior given in (2.6.9) is improper. With the choice of $\tau = 0.001$, it is expected that the resulting posterior is essentially flat.

We first implement the original Gibbs sampler, which requires the following steps:

Step 1. Sample $\boldsymbol{\beta}$ from

$$\boldsymbol{\beta}|\boldsymbol{z}, \gamma_2 \ \sim \ N_2(\hat{\boldsymbol{\beta}}, B^{-1}),$$

where $B = \tau I_2 + X'X$ and $\hat{\boldsymbol{\beta}} = B^{-1}X'\boldsymbol{z}$.

Step 2. Sample z_i from

$$z_i \ \sim \ N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1), \ \ \gamma_{y_i - 1} \leq z_i \leq \gamma_{y_i}.$$

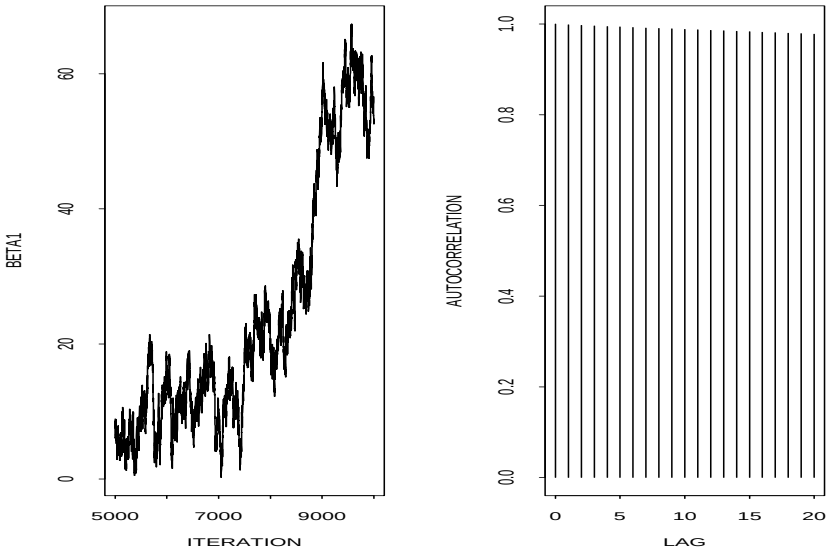

FIGURE 2.1. The original Gibbs sampler sequence of β_1 and its autocorrelation plot.

Step 3. Sample γ_2 from

$$\gamma_2|\boldsymbol{\beta}, \boldsymbol{z} \sim U[a_2, b_2],$$

where $a_2 = \max\left\{0, \max_{y_i=2} z_i\right\}$ and $b_2 = \min_{y_i=3} z_i$.

The trajectory and autocorrelation plots are displayed in Figure 2.1. These plots suggest that the original Gibbs sampler performs very poorly.

To improve the original Gibbs sampling algorithm, we consider the GM-MGMC algorithm. The group transformations proposed by Liu and Sabatti (1998) are

$$g(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z}) = (g\boldsymbol{\beta}, g\gamma_2, g\boldsymbol{z})$$

with $g > 0$. Since the Jacobian $J_g = g^{8+2+1}$ and the Haar measure $H(dg) = dg/g$, the distribution of g is

$$\pi(g|\boldsymbol{\beta}, \gamma_2, \boldsymbol{z})H(dg) \propto g^{10} \exp\{-\tfrac{1}{2}g^2[\tau\boldsymbol{\beta}'\boldsymbol{\beta} + (\boldsymbol{z} - X\boldsymbol{\beta})'(\boldsymbol{z} - X\boldsymbol{\beta})]\}.$$

In addition to the original Gibbs steps, the GM-MGMC algorithm requires the following GM step:

GM Step. Draw the group element g from $\pi(g|\boldsymbol{\beta}, \gamma_2, \boldsymbol{z})H(dg)$ by taking $g = \sqrt{g^2}$, where

$$g^2|\boldsymbol{z}, \boldsymbol{\beta} \sim \mathcal{G}(\tfrac{11}{2}, \tfrac{1}{2}[(\boldsymbol{z} - X\boldsymbol{\beta})'(\boldsymbol{z} - X\boldsymbol{\beta}) + \tau\boldsymbol{\beta}'\boldsymbol{\beta}]),$$

where $\mathcal{G}(\xi, \eta)$ denotes the gamma distribution, whose density is given by

$$\pi(g^2|\xi, \eta) \propto (g^2)^{\xi-1} \exp(-\eta g^2),$$

and *adjust* $\boldsymbol{\beta}$, γ_2, and $\boldsymbol{z}$ by

$$\boldsymbol{\beta} \leftarrow g\boldsymbol{\beta}, \quad \gamma_2 \leftarrow g\gamma_2, \quad \text{and} \quad \boldsymbol{z} \leftarrow g\boldsymbol{z}.$$

The GM-MGMC algorithm has a statistical interpretation in terms of the CA-MCMC of Liu (1998). Given fixed cutpoints, the model reduces to the probit model with the corresponding variance parameter of latent variables fixed at one. The basic idea is to expand this hidden variance by redrawing the following sufficient statistic:

$$S^2 = \sum_{i=1}^{8} (z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2.$$

To make use of the CA-MCMC algorithm, we consider the following one-to-one mapping:

$$s = S = \sqrt{S^2}, \quad e_i = (z_i - \boldsymbol{x}_i'\boldsymbol{\beta})/S, \quad \eta = \gamma_2/S, \quad \text{and} \quad \boldsymbol{\xi} = \boldsymbol{\beta}/S,$$

with the constraint $\sum_{i=1}^{8} e_i^2 = 1$. Since the Jacobian of this transformation (with fixed $\gamma_1 = 0$) is

$$J_{(z_1,\ldots,z_7,\boldsymbol{\beta},\gamma_2,z_8)\to(e_1,\ldots,e_7,\boldsymbol{\xi},\eta,s)} = s^{10}/\sqrt{e_8^2},$$

given $(e_1, ..., e_8, \eta, \boldsymbol{\xi})$ the conditional distribution of s^2 is a gamma distribution:

$$\mathcal{G}(\tfrac{11}{2}, [1 + \tau\boldsymbol{\xi}'\boldsymbol{\xi}]/2).$$

Thus, in addition to the original Gibbs steps, the CA-MCMC algorithm requires the following CA step:

CA Step. Draw s^2 from $\mathcal{G}\left(\frac{11}{2}, [1 + \tau\boldsymbol{\xi}'\boldsymbol{\xi}]/2\right)$ and *adjust* $(\boldsymbol{z}, \boldsymbol{\beta}, \gamma_2)$ by

$$(\boldsymbol{z}, \boldsymbol{\beta}, \gamma_2) \leftarrow (s/S)(\boldsymbol{z}, \boldsymbol{\beta}, \gamma_2),$$

where $S^2 = \sum_{i=1}^{8} (z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2$.

This version of CA-MCMC leads to the same result as that of the GM-MGMC algorithm.

The trajectory and autocorrelation plots of the GM-MGMC algorithm are displayed in Figure 2.2. From these plots, it is clear that the GM-MGMC algorithm substantially improves the original Gibbs sampler. However, the autocorrelations are still large. For example, the autocorrelation of β_1 at lag 10 is 0.404. This may be mainly due to the lack of information to estimate β_1, the regression coefficient for gender, which results in slow convergence of $\mu = \beta_0 + \beta_1$. To speed up the GM-MGMC algorithm further, we add another CA step that draws the parameter $\mu = \beta_0 + \beta_1$ jointly with its

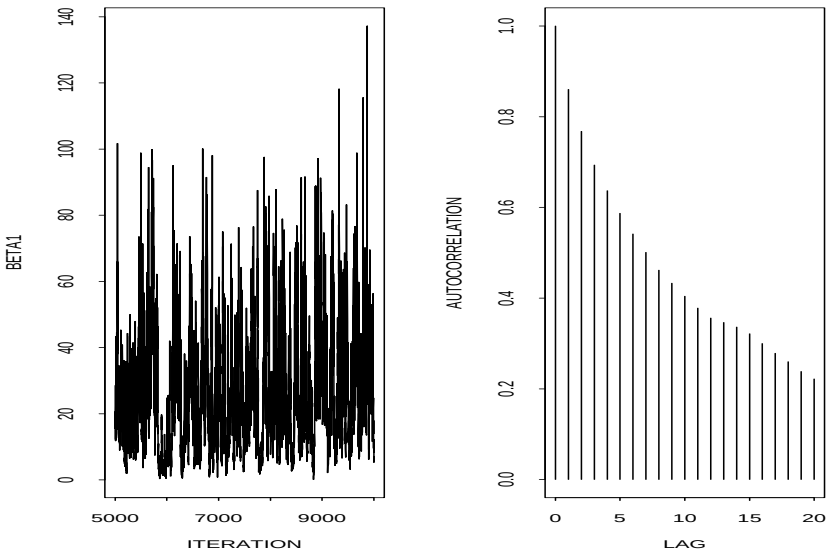

FIGURE 2.2. The GM-MGMC sequence of β_1 and its autocorrelation plot.

sufficient statistic $T = \frac{1}{4}\sum_{x_i=1} z_i$, conditioning on the current draws of $\{z_i : x_i = 0\}$, γ_2, β_0, and $\{z_i^* = z_i - T : x_i = 1\}$. Since the conditional distribution of μ given β_0 from the prior distribution of β is $N(\beta_0, 1/\tau)$, the conditional posterior distribution of (μ, T) given $\{z_i : x_i = 0\}$, γ_2, β_0, and $\{z_i^* = z_i - T : x_i = 1\}$ is

$$N_2\left(\begin{bmatrix}\beta_0\\ \beta_0\end{bmatrix}, \begin{bmatrix}1/\tau & 1/\tau\\ 1/\tau & 1/\tau + \frac{1}{4}\end{bmatrix}\right),$$

where $\max_{x_i=1}(\gamma_2 - z_i^*) \le T < \infty$. Thus, the corresponding CA step in this CA GM-MGMC algorithm can be accomplished by: (i) drawing T from

$$N(\beta_0, 1/\tau + \tfrac{1}{4}),$$

where $\max_{x_i=1}(\gamma_2 - z_i^*) \le T < \infty$, then drawing μ from

$$N\left(\beta_0 + \frac{(1/\tau)}{1/\tau + \frac{1}{4}}(T - \beta_0), \frac{1}{\tau} - \frac{(1/\tau^2)}{1/\tau + \frac{1}{4}}\right),$$

and (ii) adjusting β_1 and $\{z_i : x_i = 1\}$ by

$$\beta_1 \leftarrow \mu - \beta_0 \text{ and } \{z_i : x_i = 1\} \leftarrow \{z_i^* + T : x_i = 1\}.$$

Figure 2.3 indicates that the autocorrelations of β_1 from the CA GM-MGMC algorithm disappear even at lag 1. This simple example illustrates three important points:

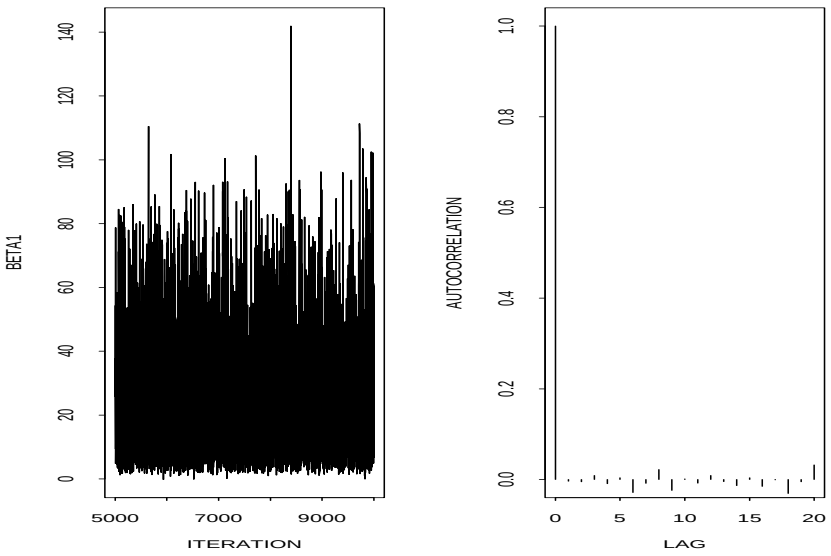

FIGURE 2.3. The CA GM-MGMC sequence of β_1 and its autocorrelation plot.

(i) the adjustment steps can dramatically improve convergence of an MCMC algorithm;

(ii) accelerating an MCMC algorithm is a continuous process; and

(iii) conditioning on sufficient statistics may play a key role in accelerating an MCMC algorithm.

2.7 Dynamic Weighting Algorithm

The *dynamic weighting method* is first introduced by Wong and Liang (1997) and further examined by Liu, Liang, and Wong (1998b). As pointed out by Liu, Liang, and Wong (1998b), the method extends the basic Markov chain equilibrium concept of Metropolis et al. (1953) to a more general weighted equilibrium of a Markov chain. The basic idea of dynamic weighting is to augment the original sample space by a positive scalar w, called a weight function, which can automatically adjust its own value to help the sampler move more freely.

Introducing the importance weights into the dynamic MC process helps make large transitions which are not allowed by the standard Metropolis transition rules. When the distribution has regions of "high" density separated by barriers of very "low" density, for example, when the tar-

get distribution is multimodal, the waiting time for a Metropolis process to cross over the barriers will be essentially infinite. In the dynamically weighted Monte Carlo, the process can often move against very steep probability barriers, which apparently violates the Metropolis rule. The weight variable is updated in a way that allows for an adjustment of the bias induced by such non-Metropolis moves.

Similar to the Metropolis algorithm, dynamic weighting starts with an arbitrary Markov transition kernel $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ from which the next candidate move is suggested. Suppose the current state is $(\boldsymbol{\theta}, w)$. Liu, Liang, and Wong (1998b) propose two dynamic weighting moves, called the Q-type move and the R-type move. Assume that the target distribution is $\pi(\boldsymbol{\theta}|D)$. Then, these two dynamic weighting schemes are given as follows:

Q-Type Move

Step 1. (Candidate state.) Draw $\boldsymbol{\vartheta} \sim T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$, and compute the Metropolis ratio

$$r(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \frac{\pi(\boldsymbol{\vartheta}|D)T(\boldsymbol{\vartheta}, \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|D)T(\boldsymbol{\theta}, \boldsymbol{\vartheta})}.$$

Step 2. (Move?) Choose $\alpha = \alpha(w, \boldsymbol{\theta}) > 0$ and draw $u \sim U(0, 1)$. Update $(\boldsymbol{\theta}, w)$ to $(\boldsymbol{\theta}^*, w^*)$ as

$$(\boldsymbol{\theta}^*, w^*) = \begin{cases} (\boldsymbol{\vartheta}, \max\{\boldsymbol{\alpha}, wr(\boldsymbol{\theta}, \boldsymbol{\vartheta})\}) & \text{if } u \le \min\{1, wr(\boldsymbol{\theta}, \boldsymbol{\vartheta})/\alpha\}, \\ (\boldsymbol{\theta}, aw) & \text{otherwise,} \end{cases} \tag{2.7.1}$$

where $a > 1$ can be either a constant or an independent random variable.

R-Type Move

Step 1. (Candidate state.) The same as the Q-type move.

Step 2. (Move?) Choose $\alpha = \alpha(w, \boldsymbol{\theta}) > 0$ and draw $u \sim U(0, 1)$. Update $(\boldsymbol{\theta}, w)$ to $(\boldsymbol{\theta}^*, w^*)$ as

$$(\boldsymbol{\theta}^*, w^*) = \begin{cases} (\boldsymbol{\vartheta}, wr(\boldsymbol{\theta}, \boldsymbol{\vartheta}) + \alpha) & \text{if } u \le \dfrac{wr(\boldsymbol{\theta}, \boldsymbol{\vartheta})}{wr(\boldsymbol{\theta}, \boldsymbol{\vartheta}) + \alpha}, \\ (\boldsymbol{\theta}, w(wr(\boldsymbol{\theta}, \boldsymbol{\vartheta}) + \alpha)/\alpha)) & \text{otherwise.} \end{cases} \tag{2.7.2}$$

For practical use of the two dynamic weighting moves, Liu, Liang, and Wong (1998b) suggest that they be applied in a compact space. This can be achieved by preventing the sampler from visiting exceedingly low-probability space. Furthermore, to guard against a possible boundary effect

caused by exceedingly small $r(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ (i.e., practically 0), one can modify the weight updating as follows: if $r(\boldsymbol{\theta}, \boldsymbol{\vartheta}) < \epsilon$ for a proposal $\boldsymbol{\vartheta}$, rejection does not induce any change of the weights.

The behavior of both dynamic weighting moves is controlled by the parameter α and the transition kernel $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$. For example, when $\alpha \to 0$, the Q-type move is identical to the R-type move, and every candidate move will be accepted. Two special cases are of great interest. First, when $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ is reversible and stationary with equilibrium distribution $\pi(\boldsymbol{\theta}|D)$, both moves reduce to the standard Metropolis algorithm. Thus, the dynamic weighting method can be viewed as an extension of the standard Metropolis algorithm. Second, when $T(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ is reversible and stationary with equilibrium distribution $g(\boldsymbol{\theta})$,

$$r(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \frac{\pi(\boldsymbol{\vartheta}|D)g(\boldsymbol{\theta})}{g(\boldsymbol{\vartheta})\pi(\boldsymbol{\theta}|D)} \quad \text{and} \quad w^* = w\frac{\omega(\boldsymbol{\vartheta})}{\omega(\boldsymbol{\theta})},$$

where $w(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|D)/g(\boldsymbol{\theta})$. Hence, if we start with $\boldsymbol{\theta}_0$ and $w_0 = c_0\omega(\boldsymbol{\theta}_0)$, then for any $i > 0$, $w_i = c_0\omega(\boldsymbol{\theta}_i)$. These weights are identical to those from the standard importance sampling method with an importance sampling distribution $g(\boldsymbol{\theta})$. Liu, Liang, and Wong (1998b) also study the behavior of the Q-type and R-type moves for several other choices of α and T.

In general, for either the Q-type move or the R-type move, the equilibrium distribution of $\boldsymbol{\theta}$ (if it exists) is not $\pi(\boldsymbol{\theta}|D)$. In this regard, Wong and Liang (1997) propose to use *invariance with respect to importance-weighting* (IWIW) as a principle for validating the above scheme and for designing new transition rules. The formal definition of IWIW is given as follows:

> *The joint distribution $\pi(\boldsymbol{\theta}, w)$ of $(\boldsymbol{\theta}, w)$ is said to be correctly weighted with respect to $\pi(\boldsymbol{\theta}|D)$ if*
>
> $$\int w\pi(\boldsymbol{\theta}, w)\, dw \propto \pi(\boldsymbol{\theta}|D). \tag{2.7.3}$$
>
> *A transition rule is said to satisfy IWIW if it maintains the correctly weighted property for the joint distribution of $(\boldsymbol{\theta}, w)$ whenever the initial joint distribution is correctly weighted.*

Suppose the starting joint distribution $\pi_1(\boldsymbol{\theta}, w)$ for $(\boldsymbol{\theta}, w)$ is correctly weighted with respect to $\pi(\boldsymbol{\theta}|D)$, i.e., $\int w\pi_1(\boldsymbol{\theta}, w)\, dw \propto \pi(\boldsymbol{\theta}|D)$. It can be shown that after a one-step transition of the R-type move with $\alpha = \alpha(\boldsymbol{\theta}, w) > 0$ for all $(\boldsymbol{\theta}, w)$, the new joint state $(\boldsymbol{\theta}^*, w^*)$ has a joint distribution $\pi_2(\boldsymbol{\theta}^*, w^*)$ that is also correctly weighted with respect to $\pi(\boldsymbol{\theta}|D)$, i.e.,

$$\int w^*\pi_2(\boldsymbol{\theta}^*, w^*)\, dw^* \propto \pi(\boldsymbol{\theta}^*|D). \tag{2.7.4}$$

If $\alpha \to 0$, then the IWIW property holds for both the Q- and R-type moves. However, when $\alpha > 0$, the Q-type move only approximately satisfies the

IWIW property. A more detailed discussion of the properties of the Q-type move can be found in Liu, Liang, and Wong (1998b).

The dynamic reweighting method has been successfully applied to simulation and global optimization problems arising from multimodal sampling, neural network training, high-dimensional integration, the Ising models (Wong and Liang 1997; Liu, Liang, and Wong 1998b), and Bayesian model selection problems (Liu and Sabatti 1999). The applications of IWIW to the computation of posterior quantities of interest will be discussed further in Chapter 3.

2.8 Toward "Black-Box" Sampling

Chen and Schmeiser (1998) propose a random-direction interior-point (RDIP) Markov chain approach to black-box sampling. The purpose of such a black-box sampler is to free the analyst from computational details without paying a large computational penalty, in contrast to specialized samplers such as the Gibbs sampler or the Metropolis–Hastings algorithm.

Assume that the target posterior distribution is of the form

$$\pi(\boldsymbol{\theta}|D) = \frac{L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta})}{c(D)}, \tag{2.8.1}$$

where $L(\boldsymbol{\theta}|D)$ is the likelihood function, $\pi(\boldsymbol{\theta})$ is a prior distribution, and $c(D)$ is the normalizing constant. We further assume that $L(\boldsymbol{\theta}|D)$ and $\pi(\boldsymbol{\theta})$ can be computed at any point $\boldsymbol{\theta}$. The key idea of RDIP is to introduce an auxiliary random variable δ so that the joint posterior distribution of $(\boldsymbol{\theta}, \delta)$ has the form

$$\pi(\boldsymbol{\theta}, \delta|D) = \begin{cases} 1/c(D), & \text{if } 0 \le \delta \le \pi(\boldsymbol{\theta}|D), \\ 0, & \text{otherwise.} \end{cases} \tag{2.8.2}$$

Integrating out δ from $\pi(\boldsymbol{\theta}, \delta|D)$ yields the marginal distribution of $\boldsymbol{\theta}$ as $\pi(\boldsymbol{\theta}|D)$. This result implies that if a Markov chain $\{(\boldsymbol{\theta}_i, \delta_i), i = 1, 2, ...\}$ has the unique uniform stationary distribution $\pi(\boldsymbol{\theta}, \delta|D)$, then the "marginal" Markov chain $\{\boldsymbol{\theta}_i, i = 1, 2, ...\}$ has a stationary distribution which is the target posterior $\pi(\boldsymbol{\theta}|D)$.

Let Ω be the interior of the $(p+1)$-dimensional region lying beneath $\pi(\boldsymbol{\theta}|D)$ and over the support of $\pi(\boldsymbol{\theta}|D)$. Then the RDIP sampler has three fundamental characteristics:

(i) Sampling generates points $(\boldsymbol{\theta}, \delta)$ from the interior of Ω.

(ii) The stationary distribution of $(\boldsymbol{\theta}, \delta)$ is uniform over Ω. Therefore, the stationary distribution of $\boldsymbol{\theta}$ is $\pi(\boldsymbol{\theta}|D)$.

(iii) The Markov chain evolution from point to point is based on random directions.

Computationally, whether $\pi(\boldsymbol{\theta}|D)$ integrates to one or not is unimportant. Suppose that $\pi(\boldsymbol{\theta}|D) = L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta})$. The advantage of this is that the normalizing constant $c(D)$ need not be computed. Based on this convention, the general version of the RDIP sampler, which essentially defines a class of samplers, can be stated as follows:

The RDIP Sampler

Step 1. (Random direction.) Generate a unit-length $(p+1)$-dimensional direction $\boldsymbol{d} \sim g_1(\boldsymbol{d}|\boldsymbol{\theta}, \delta)$.

Step 2. (Random distance.) Generate $\lambda \sim g_2(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})$.

Step 3. (Candidate point.) Set $(\boldsymbol{\theta}^*, \delta^*) = (\boldsymbol{\theta}, \delta) + \lambda \boldsymbol{d}$.

Step 4. (Candidate posterior density.) Compute $\pi^* = \pi(\boldsymbol{\theta}^*|D)$.

Step 5. (Inside Ω?) If $0 < \delta^* < \pi^*$ is false, go to Step 7.

Step 6. (Move?) Set $(\boldsymbol{\theta}, \delta) = (\boldsymbol{\theta}^*, \delta^*)$ with probability $a(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta)$.

Step 7. (Done.) Return $(\boldsymbol{\theta}, \delta)$.

The densities g_1 and g_2 in Steps 1 and 2 and the jump probability a in Step 6 must be chosen to provide a valid sampler. In typical versions, g_1, g_2, and a can be chosen so that the transition kernel of the random sequence of points $\{(\boldsymbol{\theta}_i, \delta_i),\ \ i \geq 0\}$ is doubly stochastic in Ω, guaranteeing uniformity over Ω in the limit. In general, Steps 1 and 2 can be combined to generate a $(\boldsymbol{d}, \lambda)$ conditional on $(\boldsymbol{\theta}, \delta)$. From the above description, it can be seen that the RDIP sampler is a special case of the H&R sampler.

Chen and Schmeiser (1998) discuss three variations of the general version of the RDIP sampler. One of these variations is called the state-dependent direction-and-radius sampler, which uses the location information of the current state as well as the relative height of the current location without requiring much extra computation. The detailed steps involved in this special case are given as follows:

The State-Dependent Direction-and-Radius Sampler

Step 1. (Random direction.)

- Generate a uniform unit-length p-dimensional direction $(d_1, d_2, \ldots, d_p)$;
- generate $\alpha \sim g_1^*(\alpha|r)$; and
- the $(p+1)$-dimensional random direction is $\boldsymbol{d} = (d_1 \cos\alpha, d_2 \cos\alpha, \ldots, d_p \cos\alpha, \sin\alpha)$.

Step 2. (Random distance.) Generate $\lambda \sim g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})$.

Step 3. (Candidate point.) Set $(\boldsymbol{\theta}^*, \delta^*) = (\boldsymbol{\theta}, \delta) + \lambda \boldsymbol{d}$.

Step 4. (Candidate posterior density.) Compute $\pi^* = \pi(\boldsymbol{\theta}^*|D)$ and $r^* = \delta^*/\pi(\boldsymbol{\theta}^*|D)$.

Step 5. (Inside Ω?) If $0 < \delta^* < \pi^*$ is false, go to Step 7.

Step 6. (Move?) Let $\pi(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta)$ be the conditional probability density function of the next candidate point $(\boldsymbol{\theta}^*, \delta^*)$ given the current point $(\boldsymbol{\theta}, \delta)$, and let $\pi(\boldsymbol{\theta}, \delta|\boldsymbol{\theta}^*, \delta^*)$ be the conditional probability density function by switching the positions of $(\boldsymbol{\theta}^*, \delta^*)$ and $(\boldsymbol{\theta}, \delta)$. Then set $(\boldsymbol{\theta}, \delta) = (\boldsymbol{\theta}^*, \delta^*)$ with probability $\min\{\pi(\boldsymbol{\theta}, \delta|\boldsymbol{\theta}^*, \delta^*)/\pi(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta), 1\}$.

Step 7. (Done.) Return $(\boldsymbol{\theta}, \delta)$.

Next, we discuss some possible choices of the angle density $g_1^*(\alpha|r)$, the distance density $g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})$, the conditional density $\pi(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta)$, and the jump probability.

The angle α is with respect to the horizontal plane $\Delta = \delta$. The domain of α is $(-\pi/2, \pi/2)$, with $\alpha = -\pi/2$ corresponding to a move straight down and $\alpha = \pi/2$ corresponding to a move straight up in $p+1$ dimensions. A reasonable choice of $g_1^*(\alpha|r)$ might be a mixture of beta densities

$$\begin{aligned} g_1^*(\alpha|r) =& P(\text{neg})p(\alpha|\text{neg}) + P(\text{pos})p(\alpha|\text{pos}) \\ =& r\,\frac{(-\alpha)^{a_r-1}((\pi/2)+\alpha)^{b_r-1}}{B(a_r,b_r)(\pi/2)^{a_r+b_r-1}} 1\{-\pi/2 < \alpha < 0\} \\ &+ (1-r)\,\frac{(\alpha)^{a_r-1}((\pi/2)-\alpha)^{b_r-1}}{B(a_r,b_r)(\pi/2)^{a_r+b_r-1}} 1\{0 < \alpha < \pi/2\}, \end{aligned} \tag{2.8.3}$$

where $a_r > 0$, $b_r > 0$, a_r and b_r might depend on r, and $B(a_r, b_r)$ is the beta function. Note that $E(\alpha|r) = [a_r/(a_r+b_r)](\pi/2)(1-2r)$. Here we choose the probability of moving down to be $P(\text{neg}) = r$ and the probability of moving up to be $P(\text{pos}) = 1-r$. The reason for this choice is that if the point $(\boldsymbol{\theta}, \delta)$ is close to the surface $\delta = \pi(\boldsymbol{\theta}|D)$ (i.e., r is close to 1), more probability should be assigned to negative values of α (i.e., the next move should be down), and vice versa. In addition, when r is close to 1, more probability should be assigned to the large absolute value of α, and, therefore, a_r should be large while b_r should be small. Similarly, when r is close to zero, a_r should be small and b_r should be large. Chen and Schmeiser (1998) empirically show that despite choosing a_r and b_r to be constants, the sampler still performs reasonably well.

It is desirable to choose the distance density g_2^* so that it depends upon the current location and the angle α, without incurring expensive computation. One source of almost-free information is to compute the intersection of the line through the point $(\boldsymbol{\theta}, \delta)$ with the direction $\boldsymbol{d}$ and the horizontal plane $\delta = 0$. This intersection is

$$Pt(\boldsymbol{\theta}, \boldsymbol{d}, 0) = (\boldsymbol{\theta} - (\delta \cos\alpha/\sin\alpha)(d_1, d_2, \ldots, d_p), 0)\,.$$

Similarly, the intersection of the line through the point $(\boldsymbol{\theta}, \delta)$ with the direction $\boldsymbol{d}$ and the horizontal plane $\delta = \pi(\boldsymbol{\theta}|D)$ is

$$Pt(\boldsymbol{\theta}, \boldsymbol{d}, \pi(\boldsymbol{\theta}|D)) = (\boldsymbol{\theta} - ((\delta - \pi(\boldsymbol{\theta}|D))\cos\alpha/\sin\alpha)(d_1, d_2, \ldots, d_p), \pi(\boldsymbol{\theta}|D)).$$

Then it is easy to compute the distances from the point $(\boldsymbol{\theta}, \delta)$ to $Pt(\boldsymbol{\theta}, \boldsymbol{d}, 0)$ and $Pt(\boldsymbol{\theta}, \boldsymbol{d}, \pi(\boldsymbol{\theta}|D))$, say, λ_1 and λ_2, respectively:

$$\lambda_1 = ||(\boldsymbol{\theta}, \delta) - Pt(\boldsymbol{\theta}, \boldsymbol{d}, 0)|| = \frac{\delta}{|\sin\alpha|}, \tag{2.8.4}$$

$$\lambda_2 = ||(\boldsymbol{\theta}, \delta) - Pt(\boldsymbol{\theta}, \boldsymbol{d}, \pi(\boldsymbol{\theta}|D))|| = \frac{\pi(\boldsymbol{\theta}|D) - \delta}{|\sin\alpha|}. \tag{2.8.5}$$

The distance distribution is chosen based on this information. For example, a gamma distribution might be appropriate when α is positive and a uniform distribution over $(0, \lambda_1)$ might be appropriate when α is negative. More specifically, we can choose

$$g_2^*(\lambda|\theta, \delta, \boldsymbol{d}) = g_{2a}^*(\lambda|\boldsymbol{\theta}, \delta)1\{\alpha < 0\} + g_{2b}^*(\lambda|\boldsymbol{\theta}, \delta)1\{\alpha > 0\}, \tag{2.8.6}$$

where

$$g_{2a}^*(\lambda|\boldsymbol{\theta}, \delta) = \begin{cases} |\sin\alpha|/\delta & \text{for } 0 \le \lambda \le \delta/|\sin\alpha|, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$g_{2b}^*(\lambda|\boldsymbol{\theta}, \delta) = \begin{cases} \dfrac{\lambda^2 \exp\{-6|\sin\alpha|\lambda/(\pi(\boldsymbol{\theta}|D) - \delta)\}}{\Gamma(3)\left((\pi(\boldsymbol{\theta}|D) - \delta)/6|\sin\alpha|\right)^3} & \text{for } \lambda > 0, \\ 0 & \text{otherwise.} \end{cases}$$

That is, $g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})$ in (2.8.6) is either the uniform distribution $U(0, \delta/|\sin\alpha|)$ or the gamma distribution with a shape parameter 3 and a scale parameter $(\pi(\boldsymbol{\theta}|D) - \delta)/(6|\sin\alpha|)$.

With the above choices of $g_1^*(\alpha|r)$ and $g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})$, the density of the next candidate point $(\boldsymbol{\theta}^*, \delta^*)$ conditional on the current point $(\boldsymbol{\theta}, \delta)$ is

$$\pi(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta) \propto \frac{g_1^*(\alpha|r)g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})}{|\lambda|^p|\cos\alpha|^{p-1}}, \tag{2.8.7}$$

where $\lambda = ||(\boldsymbol{\theta}^* - \boldsymbol{\theta}, \delta^* - \delta)||$, $\boldsymbol{d} = (\boldsymbol{\theta}^* - \boldsymbol{\theta}, \delta^* - \delta)/\lambda$, $\alpha = \sin^{-1}((\delta^* - \delta)/\lambda)$, and $r = \delta/\pi(\boldsymbol{\theta}|D)$. Let α^*, $\boldsymbol{d}^*$, and λ^* denote the angle, the direction, and the distance from the point $(\boldsymbol{\theta}^*, \delta^*)$ back to the point $(\boldsymbol{\theta}, \delta)$. Then $\lambda^* = ||(\boldsymbol{\theta} - \boldsymbol{\theta}^*, \delta - \delta^*)|| = \lambda$, $\boldsymbol{d}^* = (\boldsymbol{\theta} - \boldsymbol{\theta}^*, \delta - \delta^*)/\lambda = -\boldsymbol{d}$, $\alpha^* = \sin^{-1}((\delta - \delta^*)/\lambda) = -\alpha$, and $r^* = \delta^*/\pi(\boldsymbol{\theta}^*|D)$. Therefore, the jump ratio is

$$\frac{\pi(\boldsymbol{\theta}, \delta|\boldsymbol{\theta}^*, \delta^*)}{\pi(\boldsymbol{\theta}^*, \delta^*|\boldsymbol{\theta}, \delta)} = \left[\frac{g_1^*(-\alpha|r^*)}{g_1^*(\alpha|r)}\right] \cdot \left[\frac{g_2^*(\lambda|\boldsymbol{\theta}^*, \delta^*, -\boldsymbol{d})}{g_2^*(\lambda|\boldsymbol{\theta}, \delta, \boldsymbol{d})}\right]. \tag{2.8.8}$$

With $g_2^*(\lambda|\boldsymbol{\theta},\delta,\boldsymbol{d})$ given by (2.8.6), (2.8.8) can be further simplified as

$$\frac{\pi(\boldsymbol{\theta},\delta|\boldsymbol{\theta}^*,\delta^*)}{\pi(\boldsymbol{\theta}^*,\delta^*|\boldsymbol{\theta},\delta)} = \begin{cases} \dfrac{\pi(\boldsymbol{\theta}|D)}{3\pi(\boldsymbol{\theta}^*|D)} \cdot \left(\dfrac{\pi(\boldsymbol{\theta}|D)-\delta}{6|\sin\alpha|\lambda}\right)^2 \cdot \exp\left\{\dfrac{6|\sin\alpha|\lambda}{\pi(\boldsymbol{\theta}|D)-\delta}\right\} & \text{for } \alpha > 0, \\ \dfrac{3\pi(\boldsymbol{\theta}|D)}{\pi(\boldsymbol{\theta}^*|D)} \cdot \left(\dfrac{6|\sin\alpha|\lambda}{\pi(\boldsymbol{\theta}^*|D)-\delta^*}\right)^2 \cdot \exp\left\{-\dfrac{6|\sin\alpha|\lambda}{\pi(\boldsymbol{\theta}^*|D)-\delta^*}\right\} & \text{for } \alpha < 0. \end{cases} \tag{2.8.9}$$

In Step 1, the uniform distribution for a unit-length p- dimensional direction $(d_1, d_2, \ldots, d_p)$ can be extended to any continuous distribution over the surface of the p-dimensional unit sphere. Additional discussion can be found in Chen and Schmeiser (1996). Chen and Schmeiser (1998) show that with the above choices of $g_1^*(\alpha|r)$ and $g_2^*(\lambda|\boldsymbol{\theta},\delta,\boldsymbol{d})$, the Markov chain $\{(\boldsymbol{\theta}_i,\delta_i), i = 1,2,\ldots\}$ induced by the state-dependent direction-and-radius RDIP sampler is irreducible and doubly stochastic, and, therefore, has a unique stationary distribution $\pi(\boldsymbol{\theta},\delta|D)$. They further empirically study the performance of the RDIP sampler and find that the RDIP sampler works reasonably well for a bimodal distribution as well as for the ordinal response model in Section 2.5.3. The RDIP sampler is the first step toward black-box sampling. Further research in this direction needs to be done in the future.

2.9 Convergence Diagnostics

Convergence diagnostics are one of the most important components in MCMC sampling. For most practical problems, the MCMC sample generated from a user's selected MCMC sampling algorithm will ultimately be used for computing posterior quantities of interest. Thus, if a Markov chain induced by the MCMC algorithm fails to converge, the resulting posterior estimates will be biased and unreliable. As a consequence, an incorrect Bayesian data analysis will be performed and false conclusions may be drawn. Fortunately, many useful diagnostic tools along with their sound theoretical foundations have been developed during the last decade. Although no single diagnostic procedure can guarantee to diagnose convergence successfully, combining several diagnostic tools together may enable us to detect how fast or how slow a Markov chain converges and how well or how poorly a chain is mixing.

By now, the literature on convergence diagnostics is very rich. Excellent and comprehensive reviews are given by Cowles and Carlin (1996), Brooks and Roberts (1998), Mengersen, Robert, and Guihenneuc–Jouyaux (1998), and many other references therein. More recently, Robert (1998, Chap. 2) presents several useful methods on convergence control of MCMC

algorithms. In this section, we present several commonly used convergence diagnostic techniques, and we refer the reader to the above review articles for details of other available convergence diagnostic methods.

A simple but effective diagnostic tool is the trace plot. Two kinds of trace plots are useful, which are the trace plot of a single long-run sequence and the trace plots of several short-run sequences with overdispersed starting (initial) points. As discussed in Mengersen, Robert, and Guihenneuc–Jouyaux (1998), there is widespread debate about single run and multiple runs. A single sequence which has difficulty leaving the neighborhood of an attractive mode will exhibit acceptable behavior even though it has failed to explore the whole support of the target distribution $\pi(\boldsymbol{\theta}|D)$. Multiple sequences may have better exploratory power, but depend highly on the choice of starting points. On the other hand, a long-run single sequence may be advantageous in exploring potential coding bugs and the mixing behavior of the Markov chain, while multiple sequences suffer from a large increase in the number of wasted burn-in simulations for estimating posterior quantities. As a practical guideline, we suggest the use of both types of trace plots in exploring convergence and mixing behavior of the chain, and then generate a single long-run sequence with a large number of iterations (say 50,000) for estimation purposes.

For many practical problems, the dimension of the parameter space is high. Thus it may not be feasible to examine the trace plots for all parameters. In this case, we may construct trace plots for several selected parameters, which should include parameters, that are known to converge slowly, functions of parameters of interest, and some nuisance parameters. For example, for the ordinal response models in Section 2.5.3, we may need to monitor the trace plots for the regression coefficients $\boldsymbol{\beta}$, the cutpoints $\boldsymbol{\gamma}$, and some of the latent variables z_i's (nuisance parameters). If one is interested in estimating $\xi = h(\boldsymbol{\theta})$, it may be sufficient to monitor the trace plot for ξ only. However, we note that slow convergence of the nuisance parameters may seriously affect the convergence of parameters of interest as discussed in Section 2.5.3. Another related issue is that a simple time series plot for the sequence of ξ may not be effective if ξ is a discrete variable. In this regard, we propose the use of the cumulative sum (CUSUM) plot of Yu and Mykland (1998). The CUSUM plot can be constructed as follows. Given the output $\{\xi_1, \xi_2, \ldots, \xi_n\}$, we begin by discarding the initial n_0 iterations, which we believe to correspond to the burn-in period. Then, the following algorithm describes how to produce a CUSUM plot:

The CUSUM Plot

Step 1. Calculate $\bar{\xi} = (n - n_0)^{-1} \sum_{i=n_0+1}^{n} \xi_i$.

Step 2. Calculate the CUSUM

$$S_t = \sum_{i=n_0+1}^{t} (\xi_i - \bar{\xi}) \quad \text{for } t = n_0 + 1, \ldots, n.$$

Step 3. Plot s_t against t for $t = n_0 + 1, \ldots, n$, connecting successive points by line segments.

Yu and Mykland (1998) argue that the speed with which the chain is mixing is indicated by the smoothness of the resulting CUSUM plot, so that a smooth plot indicates slow mixing, while a "hairy" plot indicates a fast mixing rate for ξ.

The autocorrelation plots are the easiest tool for quantitatively assessing the mixing behavior of a Markov chain. It is important to check not only the within-sequence autocorrelations but also the intraclass (between-sequence) autocorrelations. However, care must be taken in computing autocorrelations. Without discarding iterations corresponding to the burn-in period, the autocorrelations may be under- or over-estimated, which may reflect a false mixing behavior of the Markov chain. For purposes of autocorrelation checking, a long-run single sequence may be more beneficial compared to multiple short-run sequences, since the long-run sequence will lead to more accurate estimates of autocorrelations. The slow decay in the autocorrelation plots indicates a slow mixing. On the other hand, the autocorrelations are also useful in estimating "effective sample size" for studying the convergence rates of the estimates of posterior quantities.

One of the most popular quantitative convergence diagnostics is the variance ratio method of Gelman and Rubin (1992). Gelman and Rubin's method consists of analyzing m independent sequences to form a distributional estimate for what is known about some random variable, given the observations simulated so far. Assume that we independently simulate $m \geq 2$ sequences of length $2n$, each beginning at different starting points from an overdispersed distribution with respect to the target distribution $\pi(\boldsymbol{\theta}|D)$. We discard the first n iterations and retain only the last n. Then, for any scalar function $\xi = h(\boldsymbol{\theta})$ of interest, we calculate the variance between the m sequence means defined by

$$\frac{B}{n} = \frac{1}{m-1} \sum_{i=1}^{m} (\bar{\xi}_{i\cdot} - \bar{\xi}_{\cdot\cdot})^2,$$

where

$$\bar{\xi}_{i\cdot} = \frac{1}{n} \sum_{t=n+1}^{2n} \xi_{it}, \quad \bar{\xi}_{\cdot\cdot} = \frac{1}{m} \sum_{i=1}^{m} \bar{\xi}_{i\cdot},$$

and $\xi_{it} = h(\boldsymbol{\theta}_{it})$ is the t^{th} observation of ξ from sequence i. Then we calculate the mean of the m within-sequence variances, s_i^2, each of which

has $n-1$ degrees of freedom, given by

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2,$$

where $s_i^2 = (n-1)^{-1}\sum_{t=n+1}^{2n}(\xi_{it} - \bar{\xi}_{i.})^2$. An estimator of the posterior variance of ξ is

$$\hat{V} = \frac{n-1}{n}W + \Big(1+\frac{1}{m}\Big)\frac{B}{n},$$

which is asymptotically equivalent to W. Gelman and Rubin (1992) suggest the use of a t test, deduced from the approximation $B/W \sim \mathcal{F}(m-1, df)$, where $\mathcal{F}(m-1, df)$ denotes the F-distribution with degrees of freedom $(m-1, df)$, $df = 2V^2/\widehat{\mathrm{Var}}(V)$, and

$$\begin{aligned}\widehat{\mathrm{Var}}(V) &= \Big(\frac{n-1}{n}\Big)^2\frac{1}{m}\widehat{\mathrm{Var}}(s_i^2) + \Big(\frac{m+1}{mn}\Big)^2\frac{2}{m-1}B^2 \\ &\quad + 2\frac{(m+1)(n-1)}{mn^2}\frac{n}{m}[\widehat{\mathrm{Cov}}(s_i^2, \bar{\xi}_{i.}^2) - 2\bar{\xi}_{..}\widehat{\mathrm{Cov}}(s_i^2, \bar{\xi}_{i.})].\end{aligned}$$

Then, we monitor convergence by a *potential scale reduction* (PSR) factor, which is calculated by

$$\hat{R} = (V/W)df/(df-2).$$

A large value of R suggests that either V can be further decreased by more draws, or that further draws will increase W. A value of R close to 1 indicates that each of the m sets of n simulated observations is close to the target distribution, that is, convergence is achieved. The multivariate version of the PSR can be found in Brooks and Gelman (1998). The other quantitative convergence diagnostic methods include the spectral density diagnostic of Geweke (1992), the L^2 convergence diagnostics of Liu, Liu, and Rubin (1992), and Roberts (1994), geometric convergence bounds of Rosenthal (1995a,b) and Cowles and Rosenthal (1998), the convergence rate estimator of Garren and Smith (1995) and Raftery and Lewis (1992), and many others.

As with all statistical procedures, any convergence diagnostic technique can falsely indicate convergence when in fact it has not occurred. In particular, for slowly mixing Markov chains, convergence diagnostics are likely to be unreliable, since their conclusions will be based on output from only a small region of the state space. Therefore, it is important to emphasize that any convergence diagnostic procedure should not be unilaterally relied upon. As in Cowles and Carlin (1996), we recommend using a variety of diagnostic tools rather than any single plot or statistic, and learning as much as possible about the target posterior distribution before applying an MCMC algorithm. In addition, a careful study of the propriety of the posterior distribution is important, since an improper posterior makes Bayesian inference meaningless. Also, we recommend using the acceleration

tools described in Sections 2.5 and 2.6 as much as possible, since they can dramatically speed up an MCMC algorithm.

Exercises

2.1 Using the New Zealand apple data in Example 1.1, compute posterior estimates for β and σ^2 for the constrained multiple linear regression model with the prior specification for the model parameters given in Example 2.2, using the Gibbs sampler. Compare the results to those obtained by the classical-order restricted inference in Exercise 1.2.

2.2 Simulation Study

Construct a simulation study to examine the performance of the algorithm for sampling the correlation ρ given in Example 2.3. More specifically:

(i) generate a data set $D = \{\boldsymbol{y}_i = (y_{1i}, y_{2i})',\ i = 1, 2, \ldots, n\}$ from a bivariate normal distribution $N_2(0, \Sigma)$, where $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ with different values of n and ρ;

(ii) implement a Metropolis–Hastings algorithm to obtain a Markov chain of ρ; and

(iii) study the trace plot and autocorrelation of the chain as well as the acceptance probability of the algorithm.

Some other diagnostic tools discussed in Section 2.9 may also be applied here.

2.3 Repeat Problem 2.1 using the Hit-and-Run algorithm described in Section 2.3. Compare the performance of the Gibbs sampler and the H&R algorithm.

2.4 Prove (2.5.2).

2.5 Bayesian Analysis for Sensory Data

(a) Construct a Bayesian probit model using an improper prior for the model parameters to analyze the MRE sensory data given in Table 1.2 with storage temperature, time, and their interaction as possible covariates.

(b) Derive the posterior distribution.

(c) Implement the Albert–Chib, Cowles, Nandram–Chen, and Chen–Dey algorithms described in Section 2.5.3, and compare their performance.

(d) Compute the posterior estimates of the cutpoints and the regression coefficients.

(e) Are the Bayesian results comparable to those obtained from Exercise 1.3?

2.6 Prove that the posterior $\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\epsilon}|D)$ given in (2.5.26) is proper if $\delta_0 > 0$, $\gamma_0 > 0$, and X^* is of full rank, where X^* is a matrix induced by X and $\boldsymbol{y}$ with its t^{th} row equal to $1\{y_t > 0\}\boldsymbol{x}_t'$.

2.7 Show that the conditional posterior density $\pi(\boldsymbol{\eta}|\boldsymbol{\beta}, \sigma^2, \rho, D)$ given in (2.5.29) for the reparameterized random effects is log-concave in each component of $\boldsymbol{\eta}$.

2.8 Perform a fully Bayesian analysis for the 1994 pollen count data in Example 1.3.

(i) Implement the Gibbs sampler with hierarchical centering described in Section 2.5.4 for sampling from the reparameterized posterior $\pi(\boldsymbol{\beta}, \sigma^2, \rho, \boldsymbol{\eta}|D)$ given in (2.5.28). You may choose $\delta_0 = 0.01$ and $\gamma_0 = 0.01$.
(ii) Obtain the posterior estimates for $\boldsymbol{\beta}$, σ^2, and ρ.
(iii) Compare the Bayesian estimates with those obtained from Exercise 1.5 using the GEE approach.

2.9 A Counterexample (Liu and Sabatti 1998)
If the transition function T_θ does not satisfy (2.6.2), the target distribution π may not be preserved. Let θ take values in $\{0, 1, 2, 3, 4\}$ and suppose that the target distribution is uniform, i.e., $\pi(\theta|D) = \frac{1}{5}$. The group operation is the translation: $i * j = i + j \pmod 5$. The transition functions T_θ are, respectively, $T_0(i, j) = \frac{1}{3}$ for $|i - j| \leq 1$, $i = 1, 2, 3, 4$, $T_0(0, j) = \frac{1}{3}$ for $j = 0, 1, 4$, and $T_0(4, j) = \frac{1}{3}$ for $j = 3, 4, 0$; and $T_k(i, j) = \frac{1}{5}$ for all i, j and $k > 0$.

(a) Show that π is invariant under all T_θ with θ fixed.
(b) Show that the invariant distribution of $T_i(i, j)$ is proportional to $(3, 3, 2, 2, 3)$ instead of π.

2.10 Linear Regression Models with Censored Data
Consider the experiment of improving the lifetime of fluorescent lights (Hamada and Wu 1995). Carried out by a 2^{5-2} fractional factorial design, the experiment was conducted over a time period of 20 days, with inspection every two days. The design and the lifetime data are tabulated in Table 2.2.
Let $\boldsymbol{x}_i$ be the $p \times 1$ column vector of the factor levels, including the intercepts, A, B, C, D, E, AB, and BD, and y_i be the logarithm of the corresponding lifetime for $i = 1, 2, \ldots, n$, where $p = 8$ and $n = 2 \times 2^{5-2} = 16$. Also let $(Y_i^{(l)}, Y_i^{(r)})$ denote the observed censoring interval for y_i, i.e., $y_i \in (Y_i^{(l)}, Y_i^{(r)})$. Hamada and Wu (1995) consider the following model:

$$y_i \overset{\text{i.i.d.}}{\sim} N(\boldsymbol{x}_i'\boldsymbol{\beta}, \sigma^2), \quad i = 1, 2, \ldots, n,$$

TABLE 2.2. Design and Lifetime Data for Light Experiment.

Run	Design A	B	C	D	E	Data (no. of days)	
1	+	+	+	+	+	(14, 16)	(20, ∞)
2	+	+	−	−	−	(18, 20)	(20, ∞)
3	+	−	+	+	−	(08, 10)	(10, 12)
4	+	−	−	−	+	(18, 20)	(20, ∞)
5	−	+	+	−	+	(20, ∞)	(20, ∞)
6	−	+	−	+	−	(12, 14)	(20, ∞)
7	−	−	+	−	−	(16, 18)	(20, ∞)
8	−	−	−	−	−	(12, 14)	(14, 16)

Source: Hamada and Wu (1995).

with the prior distribution specified by

$$\sigma^2 \sim \mathcal{IG}(\nu_0, \nu_0 s_0/2) \text{ and } \beta|\sigma^2 \sim N(\beta_0, \sigma^2 I_p/\tau_0)$$

for $(\boldsymbol{\beta}, \sigma^2)$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_7)'$ is the vector of regression coefficients with β_0 corresponding to the intercept, $\mathcal{IG}$ denotes the inverse gamma distribution, i.e., $\pi(\sigma^2) \propto (\sigma^2)^{-(\nu_0/2+1)} \exp\{-\nu_0 s_0/(2\sigma^2)\}$, $\nu_0 = 1$, $s_0 = 0.01$, $\boldsymbol{\beta}_0 = (3, 0, \ldots, 0)$, I_p is the $p \times p$ identity matrix, and $\tau_0 = 0.0001$.

(a) Write the posterior distribution for $(\boldsymbol{\beta}, \sigma^2)$ based on the observed data $D_{\text{obs}} = (\{(Y_i^{(l)}, Y_i^{(r)}), \boldsymbol{x}_i\},\ i = 1, 2, \ldots, n)$.
(b) Derive an expression for the posterior distribution based on the complete-observed data $D = ((y_i, \boldsymbol{x}_i),\ i = 1, 2, \ldots, n)$.
(c) Develop an efficient MCMC algorithm for sampling from the posterior distribution.
(*Hint:* Slow convergence of the original Gibbs sampler may occur; so an improved MCMC algorithm such as the GM-MGMC or CA-MCMC algorithm may be required.)
(d) Perform a fully Bayesian analysis for the lifetime data.

2.11 To sample from the posterior distribution in (2.6.9) for the rating data given in Table 2.1, consider the GM-MGMC algorithm with the following additive group transformations:

$$g(\boldsymbol{\beta}, \gamma_2, \boldsymbol{z}) = (\beta_0, \beta_1 + g, \gamma_2, \{z_i : x_i = 0\}, \{z_i + g : x_i = 1\}).$$

(a) Write the GM step.
(b) Study the performance of this version of the GM-MGMC algorithm.

2.12 Prove (2.7.4).

2.13 (i) Explain why the value of the normalizing constant $c(D)$ is not required in the RDIP sampler.

(ii) Derive (2.8.7), (2.8.8), and (2.8.9) for the conditional density and the jump ratio for the state-dependent and direction-and-radius RDIP sampler.

2.14 In Exercise 2.5, compute Gelman and Rubin's PSR factors for all four algorithms with five ($m = 5$) independent sequences of length $2n$ for $n = 500$ and $n = 1000$, and discuss which algorithm converges faster.

Springer

http://www.springer.com/978-0-387-98935-8

Monte Carlo Methods in Bayesian Computation

Chen, M.-H.; Shao, Q.-M.; Ibrahim, J.G.

2000, XIII, 387 p., Hardcover

ISBN: 978-0-387-98935-8