3. Prove that the multistage modification of the CBMC method is proper for all d (Section 5.4.3).

4. Show that the detailed balance condition $\pi(\mathbf{x})A(\mathbf{x},\mathbf{y}) = \pi(\mathbf{y})A(\mathbf{y},\mathbf{x})$ guarantees that π is the invariant distribution of $A(\mathbf{x},\mathbf{y})$.

5. Suppose the Metropolized independence sampler (MIS) is applied to sample from $\pi(\mathbf{x})$, where $\mathbf{x}$ is defined on a finite state space and the trial distribution is $g(\mathbf{x})$.

   (a) Write down the actual transition matrix A for the MIS.

   (b) Show that the second largest eigenvalue of A is $\min_{\mathbf{x}}\{\pi(\mathbf{x})/g(\mathbf{x})\}$.

   (c) Find its corresponding eigenvector.

6. Show that the random-grid method is proper [i.e., it leaves the target distribution $\pi(\mathbf{x})$ invariant]. Implement the random-grid method to sample from a multidimensional Gaussian distribution. Study empirically how the choices of k and the grid-size distribution affect algorithmic efficiency.

7. Show that the reversible jump algorithm as described in Section 5.6 leaves π invariant.

8. Show that the R-type dynamic weighting rule satisfies the IWIW property.

9. Show that the Q-type dynamic weighting rule does not satisfy the IWIW property.

10. Implement both the random-ray and the random-grid methods to replace the griddy-Gibbs method in Example 6.1 of Ritter and Tanner (1992).

11. Prove that the MTMIS algorithm gives rise to a reversible Markov chain whose equilibrium distribution is π.

6
The Gibbs Sampler

The proposal transition $T(\mathbf{x},\mathbf{y})$ in a Metropolis sampler is often an arbitrary choice out of convenience. In many applications, the proposal is chosen to be a locally uniform move. In fact, the use of symmetric and locally uniform proposals is so prevailing that these are often referred to as "unbiased proposals" in the literature. If not subjecting to the Metropolis-Hastings rejection rule, this type of move would have led to a form of simple random walk in the configuration space $\mathcal{X}$. Although such a proposal is simple both conceptually and operationally, the performance of the resulting Metropolis algorithm is often inefficient because the proposal is too "noisy." In contrast, the conditional sampling techniques to be discussed in this and the next chapters enable a MCMC sampler to follow the local dynamics of the target distribution. A distinctive feature of these MCMC algorithms is that at each iteration, they use conditional distributions (i.e., those distributions resulting from constraining the target distribution π on certain subspaces) to construct Markov chain moves. As a consequence, no rejection is incurred at any of its sampling steps. The multipoint methods described in Section 5.5 are similar in spirit but are computationally more expensive than conditional sampling.

6.1 Gibbs Sampling Algorithms

The Gibbs sampler (Geman and Geman 1984) is a special MCMC scheme. Its most prominent feature is that the underlying Markov chain is con-

structed by composing a sequence of conditional distributions along a set of directions (often along the coordinate axis).

Suppose we can decompose the random variable into d components [i.e., $\mathbf{x} = (x_1, \ldots, x_d)$]. In the Gibbs sampler, one randomly or systematically chooses a coordinate, say x_1, and then updates it with a new sample x_1' drawn from the conditional distribution $\pi(\cdot \mid \mathbf{x}_{[-1]})$, where $\mathbf{x}_{[-A]}$ refers to $\{x_j,\ j \in A^c\}$ for any subset A of the coordinate indices. Algorithmically, we can describe two types of Gibbs sampling strategy.

Random-Scan Gibbs Sampler. Let $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots x_d^{(t)})$ for iteration t. Then, at iteration $t+1$, we conduct the following steps:

- Randomly select a coordinate i from $\{1, \ldots, d\}$ according to a given probability vector $(\alpha_1, \ldots, \alpha_d)$ [e.g., $(1/d, \ldots, 1/d)$].
- Draw $x_i^{(t+1)}$ from the conditional distribution $\pi(\cdot \mid x_{[-i]}^{(t)})$ and leave the remaining components unchanged; that is, let

$$\mathbf{x}_{[-i]}^{(t+1)} = \mathbf{x}_{[-i]}^{(t)}.$$

Systematic-Scan Gibbs Sampler. Let $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots x_d^{(t)})$. At the $t+1$ iteration:

- We draw $x_i^{(t+1)}$ from the conditional distribution

$$\pi(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)})$$

  For $i = 1, \ldots, d$.

It is easy to check that *every* single conditional update step in both the random-scan and the systematic-scan Gibbs samplers leaves π invariant. To see this point, suppose $\mathbf{x}^{(t)} \sim \pi$. Then, $\mathbf{x}_{[-i]}^{(t)}$ follows its marginal distribution under π. Thus,

$$\pi(x_i^{(t+1)} \mid \mathbf{x}_{[-i]}^{(t)}) \times \pi(\mathbf{x}_{[-i]}^{(t)}) = \pi(\mathbf{x}_{[-i]}^{(t)}, x_i^{(t+1)}),$$

which means that after one conditional update, the new configuration still follows distribution π.

Under regularity conditions, one can show that a Gibbs sampler chain converges geometrically and its convergence rate is related to how the variables correlate with each other (Liu 1991, Liu, Wong and Kong 1995, Schervish and Carlin 1992). Based on a finding that the Gibbs sampler's convergence rate is controlled by the *maximal correlation* (Section 12.6.3) between the states of two consecutive Gibbs iterations, Liu, Wong and Kong (1994) and Liu (1994a) argued that grouping (some researchers also call it

blocking) highly correlated components together (i.e., update them jointly) in the Gibbs sampler can greatly improve its efficiency. Some researchers have also shown that random scan can outperform systematic scan in terms of convergence speed (Roberts and Sahu 1997).

A simple restatement of the conditional updates used in the Gibbs sampler can be potentially useful: Each Gibbs update can be seen as a random *relocation* (we used the word "perturbation" in a Metropolis sampler) of the current state $\mathbf{x}$ along a chosen direction. For example, if the first coordinate direction is chosen, then this "relocation" can be represented as

$$(x_1, \ldots, x_d) \to (x_1 + \gamma, x_2, \ldots, x_d),$$

where γ is a random draw from an appropriate distribution. It is not difficult to show that if γ is drawn from $p(\gamma) \propto \pi(x_1 + \gamma, \mathbf{x}_{[-1]})$, then the move leaves π invariant. This view is critical in generalizing the Gibbs updates to more versatile conditional moves [e.g., Markov chain updates under a transformation group setting (Liu and Wu 1999)], which are useful for designing more efficient MCMC samplers. See Chapter 8 for more discussions.

The Gibbs sampler's popularity in statistics community stems from its extensive use of *conditional distributions* in each iteration. The data augmentation of Tanner and Wong (1987) first links the Gibbs sampling structure with statistical missing data problems and the EM algorithm (see Section A.4 of the appendix for a detailed description of the algorithm). The generality and the basic theory behind Gibbs sampler were noted by Li (1988). Gelfand and Smith (1990) further demonstrated that the conditional distributions needed in Gibbs iterations are commonly available in many Bayesian and likelihood computations.

6.2 Illustrative Examples

Consider the simulation from a bivariate Gaussian distribution. Let $\mathbf{x} = (x_1, x_2)$ and let the target distribution be

$$N\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right\}.$$

The Markov chain $\mathbf{x}^{(t)} = (x_1^{(t)}, x_2^{(t)})$ corresponding to a systematic-scan Gibbs sampler is generated as

$$\begin{aligned} x_2^{(t+1)} \mid x_1^{(t+1)} &\sim N\{\rho x_1^{(t+1)}, (1-\rho^2)\}, \\ x_1^{(t+1)} \mid x_2^{(t)} &\sim N\{\rho x_2^{(t)}, (1-\rho^2)\}. \end{aligned}$$

It is seen from simple computation that

$$\begin{pmatrix} x_1^{(t)} \\ x_2^{(t)} \end{pmatrix} \sim N\left\{\begin{pmatrix} \rho^{2t-1} x_2^{(0)} \\ \rho^{2t} x_2^{(0)} \end{pmatrix}, \begin{pmatrix} 1-\rho^{4t-2} & \rho - \rho^{4t-1} \\ \rho - \rho^{4t-1} & 1-\rho^{4t} \end{pmatrix}\right\}.$$

Thus, as $t \to 0$, the joint distribution of $(x_1^{(t)}, x_2^{(t)})$ converges to the target distribution. Furthermore, the rate of convergence is equal to the maximal correlation between $x_i^{(t)}$ and $x_i^{(t+1)}$, which is ρ^2. A more general analysis along this line is given in Chapter 12.

Another simple example is given by Casella and George (1992), in which the target distribution is

$$\pi(x, y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1}$$

for $x = 0, 1, \ldots, n$ and $0 \le y \le 1$. It is easy to see that the two necessary conditional distributions are

$$\begin{aligned} x \mid y &\sim \text{Binom}(n, y) \\ y \mid x &\sim \text{Beta}(x+\alpha, n-x+\beta) \end{aligned}$$

A Gibbs sampler iterates between the above two conditional sampling steps. Figure 6.1 shows some simulation results for $n = 20$ and $\alpha = \beta = 0.5$.

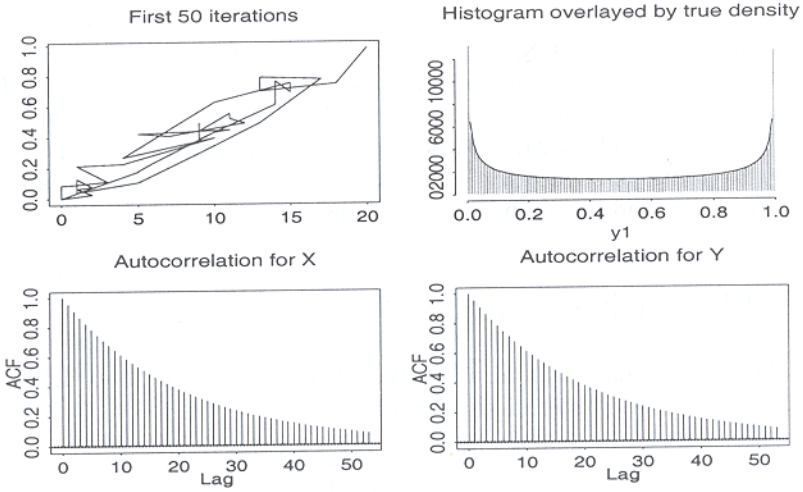

FIGURE 6.1. Using the Gibbs sampler to simulate from Beta-Binomial distribution. (a) Trace plot of first 50 iterations; (b) estimating the density of y using the Monte Carlo samples (from 200,000 iterations); (c) and (d), autocorrelation plots for x and y.

It is noted that the autocorrelation plots of x and y are almost identical — this is not an accident. It has been shown (Liu et al. 1994) that in a two-component Gibbs sampler, the two components share a common convergence rate and can be thought of as "interleaving chains" (see Chapter 12 for detailed analysis).

6.3 Some Special Samplers

6.3.1 *Slice sampler*

Suppose $\pi(\mathbf{x})$ is a density function of interest and $\mathbf{x} \in \mathbb{R}^d$. Then, drawing $\mathbf{x} \sim \pi(\mathbf{x})$ is equivalent to generating $\mathbf{z} = (z_1, \ldots, z_{d+1})$ so that it is uniformly distributed in the region S under the surface of π; that is,

$$S = \{\mathbf{z} \in \mathbb{R}^{d+1} : \; z_{d+1} \le \pi(z_1, \ldots, z_d)\}.$$

However, generating uniformly distributed random variables in an arbitrary region is equally as difficult as the original Monte Carlo simulation problem. One can apply the following Gibbs iteration to achieve the sampling:

- Draw $\mathbf{y}^{(t+1)} \sim \text{Uniform}[0, \pi(\mathbf{x}^{(t)})]$.
- Draw $\mathbf{x}^{(t+1)}$ uniformly from region $S^{(t+1)} = \{\mathbf{x} : \; \pi(\mathbf{x}) \ge \mathbf{y}^{(t+1)}\}$.

However, region $S^{(t+1)}$ in the iteration is still difficult to deal with. When π can be written as the product of k functions [i.e., $\pi(\mathbf{x}) = f_1(\mathbf{x}) \times \cdots \times f_k(\mathbf{x})$], Edwards and Sokal (1988) introduced k auxiliary variables $y_1, \ldots, y_k$ and described a Gibbs sampler for sampling $(\mathbf{x}, y_1, \ldots, y_k)$ uniformly over the region $0 < y_i < f_i(\mathbf{x}), \; i = 1, \ldots, k$:

- Draw $y_i^{(t+1)} \sim \text{Uniform}[0, f_i(\mathbf{x}^{(t)})], \; i = 1, \ldots, k$.
- Draw $\mathbf{x}^{(t+1)}$ uniformly from the region

$$S^{(t+1)} = \bigcap_{i=1}^{k} \{\mathbf{x} : \; f_i(\mathbf{x}) \ge y_i^{(t+1)}\}.$$

This method is also related to the clustering algorithms for Ising model simulations pioneered by Swendsen and Wang (1987) (Chapter 7). Damien, Wakefield and Walker (1999) showed that in many cases, one can find a decomposition of π so that the intersection set $S^{(t+1)}$ is easy to compute, which leads to an easily implemented sampler. Applications of this approach to image analysis have been discussed by Besag and Green (1993) and Higdon (1998). However, the convergence rate of the slice sampler may generally be rather slow because of the presence of many auxiliary variables.

6.3.2 *Metropolized Gibbs sampler*

When the state space of interest is discrete, Liu (1996c) suggested an "over-relaxation" strategy to improve the ordinary Gibbs sampler. Let $\mathbf{x} = (x_1, \ldots, x_d)$, where x_i takes m_i possible values, and let $\pi(\mathbf{x})$ be the distribution of interest. In the random-scan Gibbs sampler described in Section 6.1, a coordinate i is first chosen at random and the current value

x_i is replaced by a value y_i drawn from the corresponding full-conditional distribution. Here, we consider a modification of this procedure in which a value y_i, different from x_i, is drawn with probability

$$\frac{\pi(y_i \mid \mathbf{x}_{[-i]})}{1 - \pi(x_i \mid \mathbf{x}_{[-i]})},$$

then y_i replaces x_i with the Metropolis-Hastings acceptance probability,

$$\min\left\{1, \frac{1 - \pi(x_i \mid \mathbf{x}_{[-i]})}{1 - \pi(y_i \mid \mathbf{x}_{[-i]})}\right\};$$

else x_i is retained. Liu (1996c) proves that the modified Gibbs sampler for discrete random variables as defined earlier is statistically more efficient than the random-scan Gibbs sampler (see Section 13.3.1).

When $m_i = 2$, the Gibbs sampler is essentially the method of Barker (1965), whereas the modified procedure becomes a Metropolis algorithm. Peskun (1973) makes some general comparisons between these two samplers. Besag, Green, Higdon and Mengersen (1995) note that the superiority of Metropolis for binary systems results from its increased mobility around the state space. This rationale applies more generally to the Metropolized Gibbs sampler described here.

6.3.3 *Hit-and-run algorithm*

Suppose the current state is $\mathbf{x}^{(t)}$. In the hit-and-run (HR) algorithm, one does the following: (a) uniformly select a random direction $\boldsymbol{e}^{(t)}$; (b) sample a scalar $r^{(t)}$ from density $f(r) \propto \pi(\mathbf{x}^{(t)} + r\boldsymbol{e}^{(t)})$; and (c) update $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + r^{(t)}\boldsymbol{e}^{(t)}$. This HR algorithm behaves like a random-direction Gibbs sampler and allows for a complete exploration of a randomly chosen direction. It tends to be especially helpful when there are several modes (with comparable sizes) in the target distribution.

A main difficulty in implementing the HR algorithm, however, is that one is rarely able to draw from $f(r)$ in practice. Then, one may end up only using a single step of Metropolis update (Chen and Schmeiser 1993) — which renders the algorithm equivalent to the random-walk Metropolis. The MTM method introduced in Section 5.5 can help achieve the effect of conditional sampling required by the HR algorithm (Liu, Liang and Wong 2000). The following random-ray Monte Carlo scheme is a way of using MTM to achieve the hit-and-run effect. Suppose the current state is $\mathbf{x}^{(t)} = \mathbf{x}^*$; our new algorithm is as follows:

Random-Ray Monte Carlo:

- Randomly generate a direction (a unit vector) $\boldsymbol{e}$.

- Propose to draw $\mathbf{y}_1, \ldots, \mathbf{y}_k$ from a distribution $T_e(\mathbf{x}^*, \mathbf{y})$ along the direction $\boldsymbol{e}$. A generic choice is to draw i.i.d. samples $r_1, \ldots, r_k$ from $N(0, \sigma^2)$, where σ can be chosen rather large, and set $\mathbf{y}_j = \mathbf{x} + r_j\boldsymbol{e}$. Another possibility is to draw $r_j \sim \text{Uniform}(-\sigma, \sigma)$.

- Conduct the MTM step; that is, we choose $\mathbf{y}^*$ from $\mathbf{y}_1, \ldots, \mathbf{y}_k$ with probability proportional to $\pi(\mathbf{y}_j)$; and then draw $\mathbf{x}'_1, \ldots, \mathbf{x}'_{k-1}$ i.i.d. from $T_e(\mathbf{y}^*, \mathbf{x})$. Let $\mathbf{x}^* = \mathbf{x}'_k$. Then, compute the generalized Metropolis ratio

$$r = \min\left\{1, \frac{\sum_{j=1}^k \pi(\mathbf{y}_j) T_e(\mathbf{y}_j, \mathbf{x}^*)}{\sum_{j=1}^k \pi(\mathbf{x}'_j) T_e(\mathbf{x}'_j, \mathbf{y}^*)}\right\}.$$

In our experience, a much larger σ can be used compared to that in an HR with a single Metropolis update, resulting in a higher acceptance rate for the same computational time. The random-grid Monte Carlo method described in Section 5.5 can also serve as a good alternative to the HR algorithm. In Section 11.1, we will introduce another interesting variation of this algorithm — the adaptive directional sampling (Gilks, Roberts and George 1994) and conjugate gradient Monte Carlo.

6.4 Data Augmentation Algorithm

6.4.1 *Bayesian missing data problem*

A main reason for statisticians to favor conditional sampling approaches (e.g., the Gibbs sampler) over the more flexible Metropolis algorithm is that in many statistical models, it is not very difficult to derive and to sample from necessary *conditional distributions* and conditional moves are, in general, more "global" than a perturbation-type move. Another reason is that the proposal chain in the Metropolis algorithm seems to be too "random" to be effective in statistical models because the joint space of the parameter $\boldsymbol{\theta}$ and missing data $\mathbf{y}_{\text{mis}}$ in a statistical model (Section 1.9) is often "irregular." For example, θ may include a discrete component and a continuous component; or different components of $\boldsymbol{\theta}$ (such as a mean vector and a covariance matrix) may have completely different scales. For these problems, defining a reasonable "perturbation" of the current configuration of $(\boldsymbol{\theta}, \mathbf{y}_{\text{mis}})$ is very difficult. However, the Metropolis algorithm is often employed together with a conditional sampling approach (Gelman and Rubin 1992, Liu 1996a, Tierney 1994, Liu, Liang and Wong 2000).

As shown in Sections 1.9 and 3.2, we assume in a Bayesian missing data problem that the "complete-data" model $f(\mathbf{y} \mid \boldsymbol{\theta})$ has a nice analytical form from which we can do all the posterior computations in closed form. Let $\mathbf{y} = (\mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}})$, where $\mathbf{y}_{\text{obs}}$ is observed but $\mathbf{y}_{\text{mis}}$ is missing. The *observed-*

data posterior distribution is

$$p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}) = \int p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{mis}}, \mathbf{y}_{\text{obs}}) p(\mathbf{y}_{\text{mis}} \mid \mathbf{y}_{\text{obs}}) d\mathbf{y}_{\text{mis}}. \tag{6.1}$$

If we can draw $\mathbf{y}_{\text{mis}}$ from $p(\mathbf{y}_{\text{mis}} \mid \mathbf{y}_{\text{obs}})$, then a Monte Carlo approximation to (6.1) can be easily obtained. This observation forms the basis of the data augmentation algorithm.

Tanner and Wong (1987) observed that if we start with an approximation $g(\boldsymbol{\theta})$ of the target distribution, $p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}})$, we can draw m independent copies of the missing data, $\mathbf{y}_{\text{mis}}^{(1)}, \ldots, y_{\text{mis}}^{(m)}$, from

$$\tilde{p}(\mathbf{y}_{\text{mis}}) = \int p(\mathbf{y}_{\text{mis}} \mid \boldsymbol{\theta}, \mathbf{y}_{\text{obs}}) g(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{6.2}$$

This sampling can be achieved by first drawing $\boldsymbol{\theta}^{(j)} \sim g(\boldsymbol{\theta})$ and then drawing $\mathbf{y}_{\text{mis}}^{(j)} \sim p(\mathbf{y}_{\text{mis}} \mid \boldsymbol{\theta}^{(j)}, \mathbf{y}_{\text{obs}})$. The $\mathbf{y}_{\text{mis}}^{(j)}$ so produced are often called *multiple imputations* (Rubin 1987) in statistics literature. With the newly generated copies of $\mathbf{y}_{\text{mis}}$, we can form a hopefully improved approximation of the posterior distribution as

$$g_{\text{new}}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{j=1}^{m} p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}^{(j)}). \tag{6.3}$$

Note that if $g(\,)$ were indeed the true posterior distribution, (6.2) would have been the exact predictive distribution of $\mathbf{y}_{\text{mis}}$.

6.4.2 *The original DA algorithm*

A formal data augmentation scheme starts with a set of imputed missing values $\mathbf{y}_{\text{mis},1}^{(0,1)}, \ldots, \mathbf{y}_{\text{mis}}^{(0,m)}$, providing us with the first approximation of the posterior distribution: $g_0(\boldsymbol{\theta}) = m^{-1} \sum_{j=1}^{m} p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}^{(0,j)})$. Then, one carries out the following iterations.

Data Augmentation (DA) Algorithm:

- For $t = 1, \ldots N$ (N large):

  For $j = 1, \ldots, m$:

  DA1. Draw l from the set $\{1, \ldots, m\}$ at random (uniformly)

  DA2. Draw a $\boldsymbol{\theta}^*$ from $p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}^{(t-1,l)})$

  DA3. Draw $\mathbf{y}_{\text{mis}}^{(t,j)}$ from $p(\mathbf{y}_{\text{mis}} \mid \mathbf{y}_{\text{obs}}, \boldsymbol{\theta}^*)$

  End

- End

Steps DA1 and DA2 produce a sample of $\boldsymbol{\theta}$ from the mixture distribution

$$g_{t-1}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{j=1}^{m} p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}^{(t-1,j)}). \tag{6.4}$$

Thus, the random sample $\mathbf{y}_{\text{mis}}$ produced by Step DA3 follows its updated predictive distribution [i.e., (6.2) with $g(\,)$ substituted by $g_{t-1}(\,)$].

6.4.3 *Connection with the Gibbs sampler*

After a careful examination, it is observed that imputing multiple copies (m) of $\mathbf{y}_{\text{mis}}$ in each iteration is not really necessary. To illustrate this point, we look only at the first iteration. In order to produce a new imputation $\mathbf{y}_{\text{mis}}^{(1,j)}$, we need to first draw a mixture component (Step DA1), say, the one corresponding to $\mathbf{y}_{\text{mis}}^{(0,l)}$, at random, and then draw

$$\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}^{(0,l)}).$$

Effectively, we can treat $\mathbf{y}_{\text{mis}}^{(1,j)}$ as a "child" of $\mathbf{y}_{\text{mis}}^{(0,l)}$. Because of random sampling, some of the members in the zeroth generation (roughly $m/2.718$ when m is large) will have no children, which means that they will not contribute in anyway to the future approximation of the posterior distribution. In a sense, we can think that 37% of the zeroth-generation imputations are discarded (thus, wasted), completely at random. After a sufficient number of iterations, all the children will come from one ancestor (coalescence), implying that only one member in the zeroth generation contributes to the final approximation. Since the sampling of the mixture component (i.e., the parent) in producing $\boldsymbol{\theta}^*$ is completely at random, the remaining single ancestor bears no selection bias — purely by luck. Consequently, the DA procedure is mathematically equivalent to an algorithm in which one imputes only a single $\mathbf{y}_{\text{mis}}$ (i.e., $k = 1$) in each of its iterations — exactly a Gibbs sampler with two components. From now on, we will just call a two-component Gibbs sampler a *Data Augmentation Scheme*. This procedure can be illustrated more heuristically by the diagram in Figure 6.2:

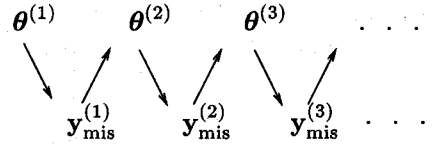

FIGURE 6.2. A graphical illustration of the data augmentation scheme.

An abstract formulation of the data augmentation approach can be summarized as follows. Suppose we are interested in simulating from a

distribution $q(\boldsymbol{\theta})$. We can construct an "augmented" system $\pi(\boldsymbol{\theta}, \mathbf{y}_{\text{mis}})$ so that the marginal distribution of $\boldsymbol{\theta}$ under this system is $q(\boldsymbol{\theta})$ [i.e., $\int \pi(\boldsymbol{\theta}, \mathbf{y}_{\text{mis}}) d\mathbf{y}_{\text{mis}} = q(\boldsymbol{\theta})$]. If this augmented system is so nice that it facilitates iterative conditional sampling, then we can simulate from it by the Gibbs sampler and obtain all necessary information regarding $q(\boldsymbol{\theta})$.

6.4.4 *An example: Hierarchical Bayes model*

We have shown in Section 1.8 that a *hierarchical Bayes* model can be used to improve predictions of students' performances from their LSAT and undergraduate GPA scores (Rubin 1980). Here, we demonstrate by a simple example how the data augmentation scheme can be used to compute with a hierarchical Bayes model.

Efron and Morris (1975) applied an empirical Bayes method to the analysis of a dataset consisting of the first 45 at-bats in the middle of a season for $n = 18$ major league players (shown in column 2 of Table 6.1). They estimated the 18 "true" batting probabilities based on this dataset, and then used them as predictions of each person's batting average for the remainder of the season. Here, we apply a hierarchical Bayes model for the same task. Let Y_i denote the observed batting average (column 2 in the table) in the first 45 at bats of the ith person, and let p_i denote his true batting percentage. A variance-stabilizing transformation of Y_i was first performed in Efron and Morris (1975): We let

$$X_i = \sqrt{45}\, \arcsin(2Y_i - 1)$$

and let

$$\theta_i = \sqrt{45}\, \arcsin(2p_i - 1).$$

Then, the X_i can be regarded as Gaussian random variables with mean θ_i and variance 1. Furthermore, a hierarchical structure is assumed on all the θ_i such that $\theta_i \sim N(\mu, \sigma^2)$ independently. Furthermore, we assume that the prior distribution for μ and σ is uniform on $(-\infty, \infty) \times (0, \infty)$, thus improper. As an exercise, the reader may try out other priors, but note that the prior for σ cannot be singular at 0.

With the model and the prior, we can implement a data augmentation scheme as follows:

- Draw θ_i, $i = 1, \ldots, 18$, conditional on μ and σ^2.
- Draw μ and σ^2 conditional on all the values of θ_i.

Figure 6.3 displays the approximated posterior density of μ estimated by Gibbs sampling (almost indistinguishable from its true posterior) as well

Player	Batting average for first 45 at-bats	Batting average for remainder	Stein's estimator	Efron-Morris's estimator
1	.400	.346	.290	.334
2	.378	.298	.286	.313
3	.356	.276	.281	.292
4	.333	.222	.277	.277
5	.311	.273	.273	.273
6	.311	.270	.273	.273
7	.289	.263	.268	.268
8	.267	.210	.264	.264
9	.244	.269	.259	.259
10	.244	.230	.259	.259
11	.222	.264	.254	.254
12	.222	.256	.254	.254
13	.222	.303	.254	.254
14	.222	.264	.254	.254
15	.222	.226	.254	.254
16	.200	.285	.249	.249
17	.178	.316	.244	.233
18	.156	.200	.239	.208

TABLE 6.1. Batting averages and their estimates.

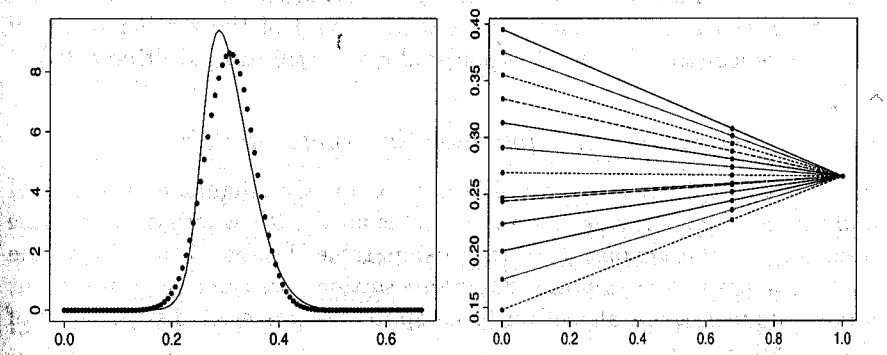

FIGURE 6.3. Left: Posterior density of μ — Gibbs sampling (solid) versus normal (dotted) approximation. Right: a graphical view of how shrinkage estimates are related to their respective MLE's.

as the shrinkage estimates.[1] In fact, this data augmentation procedure can be further modified to improve efficiency. See Liu (1994a), Gelfand, Sahu and Carlin (1995) and Liu and Sabatti (2000) for more discussions.

6.5 Finding Repetitive Motifs in Biological Sequences

In computational biology, it is often of interest to identify common patterns among a diverse class of protein or DNA sequences (Sections 1.5

[1] It is seen that the estimates of the p_i are pulled towards their common mean. In this sense, the estimate is "shrunk" toward a common point. The name "shrinkage estimator" is usually used more narrowly for estimators $\hat{p}_i$ that are shrunk towards zero.

and 4.1.3). These common patterns are usually called *"motifs"* in the literature. As illustrated in Figure 4.4, a total of K protein (or DNA) sequences, $\boldsymbol{R} = (R_1, \ldots, R_K)$, with lengths $\boldsymbol{l} = (l_1, \ldots, l_K)$ are given. They are believed to share a common motif as indicated by the blackened region; that is, every sequence in the dataset contains a subsequence of length w that is "similar" to each other. The locations of these subsequences and the motif pattern are unknown. Table 4.2 shows an alignment for a motif in multiple DNA sequences. This alignment matrix (if we know how to align the sequences) can be described by a *product-multinomial* model $\Theta = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_w]$ (Section 4.1.3). The base frequency in the background is described by an independent multinomial model $\boldsymbol{\theta}_0$.

By treating the motif locations, called the *alignment variable*, $A = \{a_1, \ldots, a_K\}$, as missing data, we can state our basic *block-motif model* more formally: Every residue/base in the dataset $\boldsymbol{R}$ outside the motif patterns (outside the blackened areas in Figure 4.4) are i.i.d. observations from Multinom($\boldsymbol{\theta}_0$) and the residue/base at position j of the motif pattern of each sequence (the blackened region) follows distribution Multinom($\boldsymbol{\theta}_j$).

6.5.1 A Gibbs sampler for detecting subtle motifs

In Section 4.1.3, we gave a uniform prior on the alignment variable A and independent Dirichlet($N_0\boldsymbol{\alpha}$) priors for all the $\boldsymbol{\theta}_j$, where $\boldsymbol{\alpha}$ is the base frequency in the genome (or other comparable database) and N_0 is the amount of pseudo-counts. Based on these settings, we can easily implement a data augmentation algorithm to compute the posterior distributions of A, $\boldsymbol{\theta}_0$, and Θ.

Data Augmentation Motif Sampler:

(a) Given a realization of the parameter values, $\boldsymbol{\theta}_0$ and Θ, we impute the alignment variable A.

(b) Given the current imputation of the alignment variable A; we sample a new realization of the parameters based on its complete-data posterior.

Step (a) can be easily achieved by "sliding" the pattern matrix Θ along each sequence and computing the relative probability of each position as the start of a motif; that is, we have

$$\pi(a_k = l \mid \Theta, \boldsymbol{\theta}_0, R_k) \propto \boldsymbol{\theta}_0^{\boldsymbol{h}(R_k)} \prod_{j=1}^{w} \left(\frac{\boldsymbol{\theta}_j}{\boldsymbol{\theta}_0}\right)^{\boldsymbol{h}(r_{k,j-1})} \propto \prod_{j=1}^{w} \left(\frac{\boldsymbol{\theta}_j}{\boldsymbol{\theta}_0}\right)^{\boldsymbol{h}(r_{k,l+j-1})}.$$

In words, the probability of $a_k = l$ is proportional to the "signal-to-noise" ratio of the sequence segment $(r_{k,l}, \ldots, r_{k,l+j-1}$. Given A, the posterior distributions of $\boldsymbol{\theta}_0$ and Θ required by Step (b) are trivial to derive provided

that their priors are standard Dirichlet or mixture Dirichlet distributions. With Dirichlet priors, these posteriors are, again, Dirichlet distributions.

It is, however, not difficult to see that given A, all the parameters can be analytically integrated out (Liu 1994a, Liu, Neuwald and Lawrence 1995), which results in a joint distribution on A:

$$\pi(A \mid \boldsymbol{R}) \propto \pi(A, \boldsymbol{R}) = \int\int \pi(\boldsymbol{R}, A \mid \boldsymbol{\theta}_0, \Theta) f(\boldsymbol{\theta}_0, \Theta) d\boldsymbol{\theta}_0 d\Theta.$$

This joint distribution can be used to derive a Gibbs sampling algorithm that focuses only on A. Although exact formulas for the conditional distributions required by the Gibbs sampler involve ratios of Gamma functions (Liu, Neuwald and Lawrence 1995), a very simple approximation, the *predictive update* form (Chen and Liu 1996), exists:

$$\pi(a_k = l \mid A_{[-k]}, \boldsymbol{R}) \propto \prod_{j=1}^{w} \left(\frac{\hat{\boldsymbol{\theta}}_{j[-k]}}{\hat{\boldsymbol{\theta}}_{0[-k]}}\right)^{\boldsymbol{h}(r_{k,l+j-1})}, \tag{6.5}$$

where $\hat{\boldsymbol{\theta}}_{j[-k]}$ is the posterior mean of $\boldsymbol{\theta}_j$ conditioned on the observation $\boldsymbol{R}$ and the current alignment $A_{[-k]}$ (excluding the kth sequence) and $\hat{\boldsymbol{\theta}}_{0[-k]}$ is the posterior mean of $\boldsymbol{\theta}_0$ based on the current non-site positions $\boldsymbol{R}_{\{A_{[-k]}\}^c}$. The formula implies that conditional on the fixed sites of the motif patterns in the remaining sequences, the probability that the motif pattern in sequence k starts at position l is proportional to the likelihood ratio of its being a motif site to its being a nonsite. Equation (6.5) forms the basis of the *site sampler* described in Section 1.5 (used in the second step for the predictive distribution). Both the exact formula and the approximation (6.5) were tested by Liu, Neuwald and Lawrence (1995) and no observable discrepancy between the two results were present.

6.5.2 Alignment and classification

It is often the case in biological applications that the sequences in consideration fall into two (or more) classes and each class has its own motif. To account for this complication, we introduce a *model variable* M: $M = 1$ stands for the one-class model and $M = 2$ for the two-class model, and assume $P(M{=}1) = P(M{=}2)$ *a priori*. One of our goals is to compute $P(M \mid \boldsymbol{R})$ (i.e., whether the data support the one-class model or prefer the two-class model). When $M{=}2$, we introduce the class indicator vector $C = (c_1, \ldots, c_K)$, with $c_i = 1$ or 2, where K is the total number of sequences. A uniform prior for C is used with the restriction that the minimal class size has to be 3. We let K_1 be the size for class one and $K_2 = K - K_1$ for class two. When $M{=}1$, we let $c_i{=}1$ for all i. Assuming

that A is independent of M *a priori*, we have

$$P(R, C, A \mid M = 2)$$
$$= \frac{\prod_{j=1}^{w}[\Gamma(\boldsymbol{h}(r_{i,a_i+j-1} : c_i = 1) + \beta)\Gamma(\boldsymbol{h}(r_{i,a_i+j-1} : c_i = 2) + \beta)]}{\Gamma(K_1 + \|\beta\|)^w \Gamma(K_2 + \|\beta\|)^w}$$
$$\times \left[\frac{\Gamma(\|\beta\|)}{\Gamma(\beta)}\right]^{2w+1} \frac{\Gamma(\boldsymbol{h}(R_{[-A]}) + \beta)}{\Gamma(\|\boldsymbol{l}\| - Kw + \|\beta\|)} \frac{1}{[\#A]} \frac{1}{2^{K-1} - \frac{K^2+K}{2} - 1}.$$

Our sampling scheme consists of the following steps:

- **Align:** For a given C, we can use the predictive updating rule (Liu, Neuwald and Lawrence 1995) to update the alignment vector A. Namely, for $\forall\, i$, we update a_i based on $P(a_i \mid C, A_{-i}, \boldsymbol{R}, M)$.
- **Fragment:** Let $A \pm 1 = \{a_1 \pm 1, \ldots, a_K \pm 1\}$; propose a move from A to $A - 1$ or $A + 1$ with equal probability and accept or reject the move based on the Metropolis ratio for $P(A \mid C, \boldsymbol{R}, M)$. This can be seen as a step of *group move*.
- **Classify:** When $M = 2$, we update C by cycling through draws from $P(c_i \mid C_{-i}, A, \boldsymbol{R}, M = 2)$, conditional on A.
- **Jump:** Conditional A, we jump between $M = 1$ and $M = 2$ based on the Metropolis ratio for $P(M, C \mid A, \boldsymbol{R})$. The proposal distribution from $M = 1$ to $(M = 2, C)$ is uniform on all allowable configuration of C. We use dynamic weighting to help the jump.

In the algorithm, a "cycle" consists of eight rounds of alignment iterations followed by one step of fragmentation and two rounds of classification iterations. The fragmentation step is a group move with the use of a translation group. This step greatly helps the convergence of alignment (Liu 1994a). After every cycle, a model jump step is conducted, with the help of a Q-type dynamic weighting (Section 5.7).

This algorithm was applied to the helix-turn-helix (HTH) dataset of Lawrence et al. (1993), which consists of 30 protein sequences with lengths ranging from 91 to 524. This set represents a large class of sequence-specific DNA binding structures involved in gene regulation. The correct locations of the motif in all the sequences were known from X-ray and nuclear magnetic resonance (NMR) structures or other experiments. The length of the motif was also determined as ~20. With w=15, our algorithm (with 2500 cycles) correctly identified all the motif locations. It provided a weighted estimate (after truncation of the weights at the 95th percentile; see Section 10.6 for details on the weight truncation method) of the posterior probability of M=2 as $\hat{p}$ <0.001.

We also applied the algorithm to another dataset consisting of the first 20 sequences in the HTH dataset and 10 new randomly shuffled sequences.

In each of the 10 random sequences, we inserted a conserved motif of length 15. The motif segment is produced from the pattern "ANHLPEQYTRGIVAK," with each position having probability 0.3 to be randomly altered. For this new dataset, the weighted estimate (truncation of the weights at the 95th percentile) of the posterior probability of M=2 is 0.94, consistent with the simulation. Conditional on $M = 2$, the algorithm (with 5000 cycles) correctly classified the sequences and correctly identified the locations of all the conserved segments. Without using dynamic weighting, the sampler induces a virtually reducible Markov chain. Acceptance probability for the reversible jump between $M = 1$ and $M = 2$ is in the range of 10^{-10}.

6.6 Covariance Structures of the Gibbs Sampler

6.6.1 Data Augmentation

Suppose $\mathbf{x} = (x_1, x_2)$ and the target distribution is $\pi(\mathbf{x})$. As we have redefined in Section 6.4.3, *data augmentation* is equivalent to a two-component Gibbs sampler; that is, with an initial value $(x_1^{(0)}, x_2^{(0)})$, data augmentation iterates as follows:

- Draw $x_1^{(t+1)}$ from the conditional distribution $\pi_{1|2}(\cdot \mid x_2^{(t)})$.
- Draw $x_2^{(t+1)}$ from the conditional distribution $\pi_{2|1}(\cdot \mid x_1^{(t+1)})$.

This sampler has some nice theoretical properties. Under some regularity conditions, it can be shown that the sampler converges geometrically and monotonically (Liu et al. 1994, Liu, Wong and Kong 1995). The convergence rate of the sampler is equal to the *maximal correlation* between the two components, which is closely related to a statistical concept, the *faction of missing information* (Rubin 1987, Liu 1994b) in Bayesian missing data problems (see Chapter 12 for more details).

From the graphical illustration in Figure 6.2, we see that $x_1^{(0)}$ is conditionally independent of $x_1^{(1)}$, given $x_2^{(0)}$. Thus, we have the following theorem.

Theorem 6.6.1 *Suppose the Markov chain resulting from a data augmentation scheme is in stationarity . Then,*

$$\text{cov}\{h(x_1^{(0)}), h(x_1^{(1)})\} = \text{var}_\pi\{E_\pi\{h(x_1) \mid x_2\}\} \tag{6.6}$$

holds for any function h.

Proof: Without loss of generality, we assume that $E_\pi h(x_1) = 0$. Then,

$$\begin{aligned}\mathrm{cov}\{h(x_1^{(0)}), h(x_1^{(1)})\} &= E\{h(x_1^{(0)})h(x_1^{(1)})\} \\ &= E[E\{h(x_1^{(0)})h(x_1^{(1)}) \mid x_2^{(0)}\}] \\ &= E[E\{h(x_1^{(0)}) \mid x_2^{(0)}\} \cdot E\{h(x_1^{(1)}) \mid x_2^{(0)}\}] \\ &= E_\pi[E_\pi\{h(x_1) \mid x_2\}]^2 \\ &= \mathrm{var}_\pi\{E_\pi\{h(x_1) \mid x_2\}\}.\end{aligned}$$

The third equation follows from the conditional independence between $x_1^{(0)}$ and $x_1^{(1)}$, given $x_2^{(0)}$; the fourth equation follows from the fact that under the stationarity assumption, both $(x_1^{(0)}, x_2^{(0)})$ and $(x_1^{(1)}, x_2^{(0)})$ follow the target distribution π. $\diamond$

More generally, an explicit expression for lag-n autocovariances can be found:

$$\mathrm{cov}[h(x_1^{(0)}), h(x_1^{(n)})] = \mathrm{var}_\pi[E_\pi[\cdots E_\pi[E_\pi\{h(x_1)|x_2\}|x_1] \mid \cdots]], \quad (6.7)$$

$$\mathrm{cov}[g(x_2^{(0)}), g(x_2^{(n)})] = \mathrm{var}_\pi[E_\pi[\cdots E_\pi[E_\pi\{g(x_2)|x_1\} \mid x_2]|\cdots]], \quad (6.8)$$

where the right-hand sides of both (6.7) and (6.8) have n expectation signs conditioned alternately on x_1 and x_2. These identities show that for a two-component Gibbs sampler, the k-lag autocovariances are non-negative and monotone nonincreasing. A similar argument can be applied to show that a random-scan Gibbs sampler also has non-negative and monotone nonincreasing autocovariances.

6.6.2 *Autocovariances for the random-scan Gibbs sampler*

Suppose $\mathbf{x}$ has d components [i.e., $\mathbf{x} = (x_1, \ldots, x_d)$]. In each iteration step of the random-scan Gibbs sampler, we independently draw an index i according to a preassigned distribution $V = (\alpha_1, \ldots, \alpha_d)$ on the index set $I = \{1, \ldots, d\}$, then replace the value of the random variable x_i corresponding to that index by a new sample drawn from the conditional distribution $\pi\{x_i \mid \mathbf{x}_{[-i]}\}$. The distribution V need not be uniform, but we do require that $\alpha_i > 0$ for all i. One can easily show that π is invariant under the above transition. It is known that the Gibbs sampler with random scanning satisfies the detailed balance relation; thus, it generates a reversible Markov chain. Besides the non-negative even-lag autocovariances guaranteed by the reversibility of the chain, Liu, Wong and Kong (1995) showed that all the autocovariances must be non-negative and monotone decreasing. Furthermore, these autocovariances can be expressed as the variances of some iterative conditional expectations. To establish these properties, we first look at the lag-1 autocovariance.

Lemma 6.6.1 *Let $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ be two consecutive realizations of the random-scan Gibbs sampler under stationarity, and let i be the random variable representing which index is updated at stage one, taking values on $I = \{1, \ldots, d\}$ with distribution V. Then, for any $h(\,) \in L_0^2(\pi)$,*

$$\begin{aligned}\mathrm{cov}\{h(\mathbf{x}^{(0)}), h(\mathbf{x}^{(1)})\} &= E\left[\sum_{i=1}^d \alpha_i E^2\{h(\mathbf{x}) \mid \mathbf{x}_{[-i]}\}\right] \\ &= E\left[E^2\{h(\mathbf{x}) \mid \mathbf{i}, \mathbf{x}_{[-\mathbf{i}]}\}\right] \geq 0.\end{aligned}$$

Proof: From the definition of the scan, it is understood that

$$\begin{aligned}E\{h(\mathbf{x}^{(0)})h(\mathbf{x}^{(1)})\} &= E[E[E\{h(\mathbf{x}^{(0)})h(\mathbf{x}^{(1)}) \mid \mathbf{i}, \mathbf{x}^{(0)}\} \mid \mathbf{x}^{(0)}]] \\ &= \sum_{i=1}^d \alpha_i E[E\{\mathbf{x}^{(0)})h(\mathbf{x}^{(1)}) \mid \mathbf{i} = i, \mathbf{x}_{[-i]}^{(0)}\}] \\ &= E\left[\sum_{i=1}^d \alpha_i E^2\{h(\mathbf{x}^{(1)}) \mid \mathbf{x}_{[-i]}^{(0)}\}\right] \\ &= E\left[E^2\{h(\mathbf{x}) \mid \mathbf{i}, \mathbf{x}_{[-\mathbf{i}]}\}\right] \geq 0.\end{aligned}$$

The second equality follows from our understanding of the random scan; the third equality is true because conditioned on a chosen updating index $\mathbf{i} = i$ and fixed values of the corresponding components, $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(1)}$ are independent and identically distributed under stationarity. $\diamond$

The lemma suggests setting α_i small when $E[E^2\{h(\mathbf{x}) \mid \mathbf{x}_{[-i]}\}]$ is large. Since $h(\mathbf{x})$ has mean zero, we also have

$$E[E^2\{h(\mathbf{x}) \mid \mathbf{x}_{[-i]}\}] = \mathrm{var}\{h(\mathbf{x})\} - E[\mathrm{var}\{h(\mathbf{x}) \mid \mathbf{x}_{[-i]}\}].$$

Hence, α_i should be set small if $E[\mathrm{var}\{h(\mathbf{x}) \mid \mathbf{x}_{[-i]}\}]$ is small, which can be understood as that one should make fewer visits to a component that is less variable.

Theorem 6.6.2 *Let $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$, be consecutive samples generated by the random scan under stationarity, and let $\mathbf{i}$ be the random variable representing the random index in the updating scheme. For $h(\mathbf{x}) \in L_0^2(\pi)$, the lag-$n$ autocovariance between $h(\mathbf{x}^{(0)})$ and $h(\mathbf{x}^{(n)})$ is a non-negative monotone decreasing function of n. It can be written as*

$$\mathrm{cov}\{h(\mathbf{x}^{(0)}), h(\mathbf{x}^{(n)})\} = \mathrm{var}[E[\cdots E[E\{h(\mathbf{x}) \mid \mathbf{i}, \mathbf{x}_{[-\mathbf{i}]}\} \mid \mathbf{x}] \mid \cdots]], \quad (6.9)$$

where there are n conditional expectations taken alternately on $\{\mathbf{i}, \mathbf{x}^{[-\mathbf{i}]}\}$ and $\mathbf{x}$.

Proof: The expression is derived by repeatedly applying Lemma 6.6.1 and the Markov property. The monotonicity is a simple property of conditional expectations. $\diamond$

In Chapter 12, we show how these expressions can be used to compare different sampling schemes and how the the *maximal correlation* among the d variables relates to the convergence rate of the scheme.

6.6.3 *More efficient use of Monte Carlo samples*

An interesting and immediate consequence of (6.7) and (6.8) is that Rao-Blackwellization *always* increases computational efficiency of Monte Carlo estimates. This problem can be formulated as follows: Suppose we are interested in estimating $I = E_\pi[h(x_1)]$ using the Monte Carlo samples obtained by data augmentation. Then, we have at least two possible estimators:

$$\hat{I} = \frac{1}{m}\left\{h(x_1^{(1)}) + \cdots + h(x_1^{(m)})\right\} \tag{6.10}$$

$$\tilde{I} = \frac{1}{m}\left\{E[h(x_1) \mid x_2^{(1)}] + \cdots + E[h(x_1) \mid x_2^{(m)}]\right\}. \tag{6.11}$$

The first estimator $\hat{I}$ is termed the *histogram estimator* and the second is called the *mixture estimator*. The name "mixture" stems from the fact that expression (6.11) is a mixture of complete-data posterior distributions (a natural choice of the kernel densities for a smooth estimate of the density curve). It has been pointed by Gelfand and Smith (1990) that the second estimation $\tilde{I}$ should be preferred — but their argument was based on the assumption that the Monte Carlo samples $\mathbf{x}^{(j)}$ are mutually independent (as explained in Section 3.4.6), which is clearly false in a Gibbs sampler. However, by using (6.7) and (6.8), we see that under stationarity,

$$m^2\mathrm{var}(\hat{I}) = m\sigma_0^2 + 2(m-1)\sigma_1^2 + \cdots + 2\sigma_{m-1}^2, \tag{6.12}$$

where $\sigma_k^2 = \mathrm{cov}[h(x_1^{(0)}), h(x_1^{(k)})]$; we have shown from (6.7) that this quantity is non-negative. By using the expression (6.8), we can derive that

$$m^2\mathrm{var}(\tilde{I}) = m\sigma_1^2 + 2(m-1)\sigma_2^2 + \cdots + 2\sigma_m^2. \tag{6.13}$$

Comparing the two variances, we see that each term in (6.13) is exactly one lag behind the corresponding term in (6.12). Because of monotonicity of the autocovariances, we conclude that $\mathrm{var}(\tilde{I}) \leq \mathrm{var}(\hat{I})$.

6.7 Collapsing and Grouping in a Gibbs Sampler

Let $\mathbf{x} = (x_1, \ldots, x_d)$ be a random variable that can be partitioned into d components, with density $\pi(\mathbf{x})$. We consider a systematic-scan Gibbs sampler being applied to sample from this target distribution; that is, a

Markov chain $\{\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_d^{(t)}), t = 0, 1, \ldots\}$ is constructed with its transition function defined by the d-component Gibbs sampler,

$$K(\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}) = \prod_{l=1}^{d} \pi\{x_l^{(t+1)} \mid x_1^{(t+1)}, \ldots, x_{l-1}^{(t+1)}, x_{l+1}^{(t)}, \ldots, x_d^{(t)}\}. \tag{6.14}$$

It is easy to check that $\pi(\mathbf{x})$ is invariant under this transition.

Suppose the last two components x_{d-1} and x_d can be drawn together; then, we have a reduced Gibbs sampler on a new partition of the random variable $\mathbf{x}^* = \{x_1, \ldots, x'_{d-1}\}$, where $x'_{d-1} = \{x_{d-1}, x_d\}$, by *grouping*. Furthermore, suppose that the component x_d can be integrated out, then an even more reduced sampler on $\mathbf{x}^- = \{x_1, \ldots, x_{d-1}\}$, with its marginal density $\pi(\mathbf{x}^-) = \int \pi(\mathbf{x})\, dx_d$, results from *collapsing*. We will compare the three schemes.

In order to argue rigorously, we introduce some concepts concerning a Markov chain and its associated function spaces. Let $L^2(\pi)$ denote the set of all functions $h(\,)$ that are square integrable with respect to π (i.e., has a finite variance). This set is a *Hilbert space* with an inner product defined by $\langle h, g\rangle = E_\pi\{h(\mathbf{x})g(\mathbf{x})\}$. Thus, $\|h\| = \mathrm{var}_\pi(h)$. Let $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ be a general state-space Markov chain with transition function $K(\mathbf{x}, \mathbf{y}) = P(\mathbf{x}^{(1)} = \mathbf{y} \mid \mathbf{x}^{(0)} = \mathbf{x})$. We define the *forward* operator $\mathbf{F}$ on $L^2(\pi)$ for the Markov chain as

$$\mathbf{F}h(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{y})h(\mathbf{y})d\mathbf{y} = E\{h(\mathbf{x}^{(1)}) \mid \mathbf{x}^{(0)} = \mathbf{x}\}.$$

We observe immediately that the *norm* of the operator is at most 1, where the norm is defined as $\|\mathbf{F}\| = \sup_h \|\mathbf{F}h(\mathbf{x})\|$ with the supremum taken over all functions with $E(h^2) = 1$. On the other hand, since the constant function c is an eigenfunction of the operator corresponding to eigenvalue 1, we know that the norm of $\mathbf{F}$ is exactly 1. When the chain is *reversible* [i.e., the *detailed balance* condition $\pi(\mathbf{x})K(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y})K(\mathbf{y}, \mathbf{x})$ is satisfied], $\mathbf{F}$ is a self-adjoint operator. When $\boldsymbol{F}$ is compact and self-adjoint, (which is true when the state space is finite and the chain is reversible), the second largest eigenvalue (in absolute value) of $\boldsymbol{F}$ characterizes the mixing rate, or convergence rate, of the Markov chain. Many methods are available for bounding the second largest eigenvalue and finding the actual rate of convergence for this case (Diaconis 1988, Diaconis and Stroock 1991). Methods for dealing with nonreversible chain also exist, although rare (Fill 1991); see Section 12.6 for more details.

Now, we consider $L_0^2(\pi) = \{h(\mathbf{x}) \in L^2(\pi) : E\{h(\mathbf{x})\} = 0\}$, which is a subspace of $L^2(\pi)$ of all mean zero functions. Clearly, this is again a Hilbert space with the same inner product and is invariant under the operator $\mathbf{F}$. We use $\mathbf{F_0}$, called the *forward operator*, to denote the operator on $L_0^2(\pi)$ induced by $\mathbf{F}$. Then, the largest eigenvalue of $\mathbf{F_0}$ is exactly the same as the second largest eigenvalue of $\mathbf{F}$. Typically, the spectral radius of $\mathbf{F_0}$

characterizes the rate of convergence of the Markov chain in both reversible and nonreversible cases. When the chain is reversible, the spectral radius of $\mathbf{F_0}$ is the same as its norm. A general relationship between the norm and the spectral radius of an operator is

$$\lim_{n\to\infty} \|\mathbf{F_0}^n\|^{1/n} = r,$$

where r is the spectral radius. This suggests that one can compare different Markov chains by comparing the norms of the corresponding forward operators. It is interesting to note here that $\|\mathbf{F_0}\|^2$ equals the second largest eigenvalue of the transition operator for the reversiblized chain, which ties in with the method of Fill (1991).

Let $\mathbf{F_s}$ denote the forward operator for the standard Gibbs sampler, corresponding to the transition function (6.14); let $\mathbf{F_g}$ be the forward operator corresponding to the grouping procedure and let $\mathbf{F_c}$ for the collapsed Gibbs sampler with x_d integrated out. The three samplers can be illustrated by the diagrams of their respective visiting schemes:

$$\begin{aligned} \mathbf{F_s}: &\quad x_1 \to x_2 \to \cdots \to x_d; \\ \mathbf{F_g}: &\quad x_1 \to x_2 \to \cdots \to \{x_{d-1}, x_d\}; \\ \mathbf{F_c}: &\quad x_1 \to x_2 \to \cdots \to x_{d-1}. \end{aligned} \tag{6.15}$$

Theorem 6.7.1 (Three-schemes theorem) *The norms of the three forward operators are ordered as*

$$\|\mathbf{F_c}\| \le \|\mathbf{F_g}\| \le \|\mathbf{F_s}\|.$$

A similar result for the random-scan Gibbs sampler is proved in Section 13.2.2. This theorem can be understood from the the diagrams in Figures 6.4 and 6.5. Consider simulating a three-component random variable $\mathbf{x} = (x_1, x_2, x_3)$ from $\pi(\mathbf{x})$. The deterministic-scan Gibbs sampler is depicted in Figure 6.4 and the samplers resulting from *grouping* and *collapsing* are shown in Figure 6.5.

To illustrate the foregoing theorem, we compared the regular data augmentation and the collapsing approach for the bivariate Gaussian problem with Murray's data (Section 4.4.1). Because the posterior distribution of the unknown covariance matrix Σ is "easy" when given completed data and, given Σ, imputing the missing data is easy, we can implement a standard data augmentation algorithm by iterating between

$$[\Sigma \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}]$$

and

$$[\mathbf{y}_{\text{mis}} \mid \mathbf{y}_{\text{obs}}, \Sigma].$$

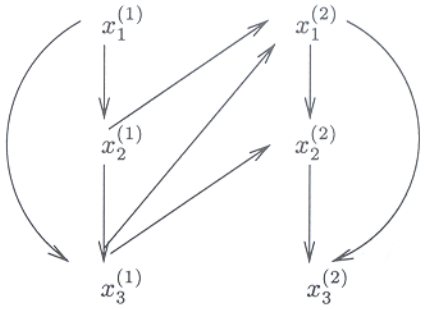

FIGURE 6.4. A graphical illustration of the standard Gibbs sampler with three components. The arrows represents the "causal relationships" in Gibbs sampling: $x_1^{(2)}$ is drawn conditional on $(x_2^{(1)}, x_3^{(1)})$; $x_2^{(2)}$ is drawn conditional on $(x_3^{(1)}, x_1^{(2)})$; and $x_3^{(2)}$ is drawn conditional on $(x_1^{(2)}, x_2^{(2)})$

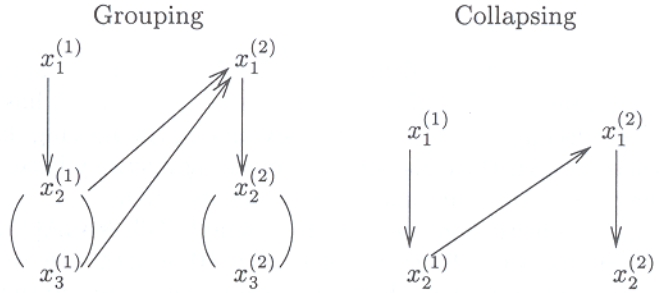

FIGURE 6.5. Graphical representations of grouping and collapsing schemes. The arrows represent the "causal relationships" in Gibbs sampling.

In a collapsing scheme, we can integrate out Σ and iterate only among the missing values; that is, we can iterate the step

$$[y_{\text{mis},i} \mid \mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis},[-i]}],$$

which is a noncentral t-distribution whose accurate form can be found in Kong et al. (1994, Section 3.1). More precisely, conditional on the current imputed values for all the missing components $y_{\text{mis},j}$, $j \neq i$, we can easily update $y_{\text{mis},i}$.

To compare the collapsed and the standard schemes, we compute autocovariance curves for each of the eight missing components. Figure 6.6 contains two groups of autocovariance curves; within each group, there are eight curves for eight missing values, respectively. They are estimated from 20,000 iterations for each chain. Since both chains are geometric mixing, we fit the model $\text{auto}(n) = C\rho^n + \epsilon$ to the autocovariances for the two bundles, respectively, where $\text{auto}(n)$ denotes the lag-n autocovariance. It

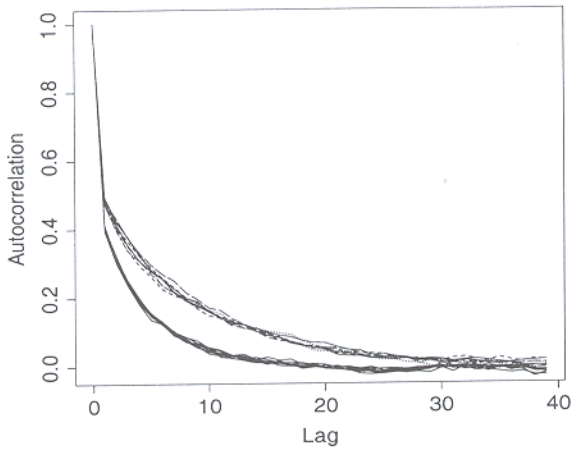

FIGURE 6.6. Autocovariance plot for both the standard and the collapsed Gibbs sampling scheme. Upper group: the standard; lower group: the collapsed.

is seen that the $\hat{\rho}$ estimated from the standard scheme is about 2.5 times larger than that of the collapsed scheme. The ordinary DA has an intuitive appeal for "decoupling" the dependency among the missing data. But our results showed that the collapsing scheme is significantly better in terms of both the convergence rate and the cost per iteration.

As another illustration, we consider a simple Gibbs sampling algorithm for the nonparametric Bayes problem in Section 4.4.2, where we describe a sequential imputation method for the posterior computation. The algorithm described here and its various improvements can be found in Escobar (1994), Liu (1996b), and MacEachern (1994). Recall that we observe $y_i \sim \text{Binom}(l_i, \zeta_i)$, and assume that $\zeta_i \overset{\text{i.i.d.}}{\sim} F$ and $F \sim \mathcal{D}(\alpha)$, where $\mathcal{D}(\alpha)$ represents a Dirichlet process. By the Bayes theorem, we obtain the predictive distribution of ζ_i [equivalent to (4.11)]:

$$\pi(\zeta_i \mid y_i, \boldsymbol{\zeta}_{[-i]}) \propto \zeta_i^{y_i}(1-\zeta_i)^{l_i-y_i}\alpha(\zeta_i) + \sum_{j\neq i} \zeta_j^{y_j}(1-\zeta_j)^{l_j-y_j}\delta_{\zeta_j}(\zeta_i).$$

Hence, a collapsed Gibbs sampler (with F collapsed down) can be applied to iteratively sample ζ_i using the foregoing predictive distribution.

Moreover, the Gibbs sampling algorithm for finding repetitive motifs in biological sequences as illustrated by (6.5) is also an application of the collapsing theorem in which the parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\Theta}$ are integrated out from the model (Liu 1994a).

The collapsing theorem suggests that one should avoid introducing unnecessary components into a Gibbs sampler. This is in agreement with the common wisdom for Monte Carlo computation: Do as much analytical work as possible. However, in the next chapter, we show that in the Ising model one can greatly improve computational efficiency by introducing a clever auxiliary variable.

6.8 Problems

1. Implement a Gibbs sampler for simulating from a Ising model defined on a 32×32 grid.

2. Write down the transition matrix of a random-scan Gibbs sampler as defined in Section 6.1. Show that the resulting Markov chain is reversible.

3. Write down the transition matrix of a systematic-scan Gibbs sampler as defined in Section 6.1. Show that the resulting Markov chain is nonreversible.

4. Evaluate the efficiency gain in using the mixture estimate versus the histogram estimate for the bivariate Gaussian example and the hierarchical Bayes example discussed in Section 6.4.4.

5. Why is the random-grid Monte Carlo method described in Section 5.5 a sensible alternative to the hit-and-run algorithm?

6. Explain why the original data augmentation algorithm of Tanner and Wong (1987) is practically equivalent to a Gibbs sampler with two components.