Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. James C. Spall
Copyright © 2003 John Wiley & Sons, Inc.,
ISBN: 0-471-33052-3

CHAPTER 16

MARKOV CHAIN MONTE CARLO

The preceding two chapters considered the interface of simulation and optimization. This chapter on Markov chain Monte Carlo (MCMC) continues the study of simulation-related methods, but with a different focus. MCMC is a powerful means for generating random samples that can be used in computing statistical estimates, numerical integrals, and marginal and joint probabilities. The approach is especially useful in statistical applications where one is forming an estimate based on a multivariate probability distribution or density function that is only available to within an unknown constant. MCMC provides a means for generating samples from *joint* distributions based on easier sampling from *conditional* distributions. The approach has had a large impact on the theory and practice of statistical modeling. In fact, MCMC sometimes applies in problems where it is hard to imagine any other approach working.

The two most popular specific implementations of MCMC are the Metropolis–Hastings (M-H) algorithm and Gibbs sampling. Over the past 10–15 years, these two implementations of MCMC have revolutionized aspects of statistical modeling and data analysis by providing a practical general framework for many problems that formerly would have required significant application-specific methodology. Gibbs sampling, in particular, has had an especially significant impact in Bayesian problems, but Gibbs sampling applies to non-Bayesian problems as well.

Section 16.1 provides some general background related to random sampling and ergodic averaging. Section 16.2 discusses the M-H algorithm and Section 16.3 discusses Gibbs sampling. Section 16.4 presents some theoretical justification for Gibbs sampling and Section 16.5 presents several numerical examples. Section 16.6 discusses the important special case of Bayesian analysis. Finally, Section 16.7 offers some summary remarks on the relative properties of M-H and Gibbs sampling.

16.1 BACKGROUND

Markov chain Monte Carlo (MCMC) is a powerful means for generating random samples that can be used in computing statistical estimates and in computing marginal and conditional probabilities. MCMC methods rely on a *dependent* (Markov) sequence with a limiting distribution corresponding to a distribution of interest.[1] This contrasts with many classical Monte Carlo methods, which are based on *independent* samples. MCMC methods apply to a broader class of multivariate problems and are frequently easier to implement.

Although MCMC has general applicability, one area where MCMC has had a revolutionary impact is Bayesian analysis. MCMC has greatly expanded the range of problems for which Bayesian methods can be applied. Metropolis et al. (1953) introduced MCMC-type methods in the area of physics. Following key papers by Hastings (1970), Geman and Geman (1984), and Tanner and Wong (1987), the paper of Gelfand and Smith (1990) is largely credited with introducing the application of MCMC methods in modern statistics, specifically in Bayesian modeling.

Over the last decade, many papers and books have been published displaying the power of MCMC in dealing with realistic problems in a wide variety of areas. Among the excellent review papers in this area are the survey by Besag et al. (1995) and the vignettes by Cappé and Robert (2000) and Gelfand (2000). The survey by Evans and Swartz (1995) puts MCMC in perspective relative to other powerful methods for numerical integration in a statistical setting. Among the many books devoted to MCMC and its close algorithmic relatives are Gilks et al. (1996), Robert and Casella (1999), Chen et al. (2000), and Liu (2001). The February 2002 issue of the *IEEE Transactions on Signal Processing* is devoted to MCMC and closely related methods, as used in signal processing and tracking applications.

The treatment here is merely a glance at this booming area of research and practice. The aim is to provide the reader with some of the central motivation and the rudiments needed for a straightforward application. Obviously, the books and other references provide extensive detail not included here. Following convention, we use the terms *density* and *distribution* interchangeably when referring to the mechanism for generating a random process; this does not imply, of course, that a density is the same as a distribution function.

The prototype problem is as follows. Suppose that there is a process generating a random vector $\boldsymbol{X}$, and we wish to compute $E[f(\boldsymbol{X})]$ for some function $f(\cdot)$. We are interested in the case where this expectation is not readily available via standard analytical means. For the moment, let us suppose that $\boldsymbol{X}$ is a continuous random vector with an associated probability density function $p(\boldsymbol{x}) = p_X(\boldsymbol{x})$. The methods to be described are, in fact, quite general and are not inherently tied to this assumption of continuity. We usually adopt this restriction for ease of presentation and because continuous random processes are very common. Hence, we may write

[1] As discussed in Appendix E, the term *Markov chain* is sometimes reserved for use with processes having discrete outcomes. In general applications of MCMC, the relevant processes may have discrete, continuous, or hybrid outcomes. For consistency with standard terminology in the MCMC area, we follow suit in this chapter in using the term *Markov chain* under the more general application to discrete or continuous outcomes.

$$E[f(X)] = \int f(x)p(x)\,dx, \tag{16.1}$$

where the integral is over the domain for X. The density $p(x)$ is sometimes called the *target density*. More generally, the target density (distribution) represents the distribution for the random variables of interest for the analysis. In some cases, for example, the target will pertain to a subset of the elements in X (e.g., it may represent the marginal distribution for only the first component of X).

A standard Monte Carlo method for approximating the integral in (16.1) is to draw n independent, identically distributed (i.i.d.) samples of X, say X_k, $k = 1, 2,\ldots, n$, from the density function $p(x)$. We then form an average based on these independent samples, leading to

$$E[f(X)] \approx \frac{1}{n}\sum_{k=1}^{n} f(X_k).$$

Because the samples X_k are i.i.d., the strong and weak laws of large numbers (Appendix C) ensure that the approximation can be made as accurate as desired with increasing n.

Often, however, drawing samples from the density $p(x)$ is not feasible. The density may be very complicated and perhaps not even available analytically. Standard random number generation methods, such as in Appendix D, are generally quite limited in the type of randomness that can be produced. In practice, many distributions are not of the restricted "named" type for which most standard methods apply. The accept–reject method, which is much more general than the inverse transform method, is usually only computationally practical if the structure of $p(x)$ is exploited to find a "tight" majorizing function (typically requiring an optimization process). A related general method for random number generation is a version of the accept–reject method called adaptive rejection sampling (Robert and Casella, 1999, pp. 56–59 and 232). This method is restricted to densities that are log-concave (i.e., the logarithm of the density is a concave function) and is known to be very inefficient in high-dimensional problems (computations proportional to the *fifth power* of dim(X); see, e.g., Gilks et al., 1996, p. 84). Moreover, this sampling method may not be as efficient as the Markov chain-based methods described below (Robert and Casella, 1999, p. 241).

Fortunately, the integral approximation above—with its i.i.d. summands and requirement for direct sampling from $p(x)$—is more restrictive than necessary to form an estimate of $E[f(X)]$. An integral can, in fact, be approximated by a possibly *dependent* sample $\{X_k\}$ that properly reflects the proportions associated with $p(x)$. This less stringent requirement opens up the possibility of efficient Markov chain-based schemes that avoid the need to directly sample from $p(x)$. In particular, one can use Markov chain-based Monte Carlo methods to efficiently produce dependent sequences having $p(x)$ as a limiting distribution *without the difficult or impossible task of sampling directly* from $p(x)$. An additional important benefit of MCMC methods is that $p(x)$ need only be known to within a scale factor. This is especially relevant in Bayesian applications, as discussed in Sections 16.2 and 16.6.

Consider a sequence $X_0, X_1, X_2, \ldots$ such that X_{k+1} is generated from the (conditional) distribution for $\{X_{k+1}|X_k\}$ and X_0 represents some initial condition (not needed in the i.i.d. case above). By the form of the conditional distribution, knowledge of X_k provides the information required to probabilistically characterize the behavior of the state X_{k+1}. That is, the distribution for X_{k+1} depends only on the current state X_k, not on the earlier states $X_0, X_1, \ldots, X_{k-1}$. Hence, $X_0, X_1, X_2, \ldots$ is a Markov chain, as discussed in Appendix E (but see footnote 1 in the current chapter on the use of the term *Markov chain*).

Under standard conditions for Markov chains, the dependence of X_k on any fixed number of early states, say $X_0, X_1, \ldots, X_M$, $M < \infty$, disappears as $k \to \infty$. Hence, the density (distribution) of X_k approaches a stationary form, say $p^*(\cdot)$. That is, as k gets large, the (dependent) random vectors in the Markov sequence have a common distribution with, say, density $p^*(\cdot)$. Ignoring the first M iterations in the chain (called the *burn-in period*), we can form an *ergodic average*

$$\frac{1}{n-M}\sum_{k=M+1}^{n} f(X_k), \qquad (16.2)$$

so called because it is a practical realization of the famous ergodic theorem of stochastic processes.

The ergodic theorem guarantees that the normalized sum in (16.2) will approach the mean of $f(X)$ (usually in the mean square [m.s.] or almost sure [a.s.] sense; see Appendix C) as $n \to \infty$ for any fixed M, where this mean is computed with respect to $p^*(\cdot)$. For the ergodic theorem for m.s. convergence to hold, it is necessary and sufficient that: (i) $\mathrm{cov}[f(X_j), f(X_k)]$ is uniformly bounded in magnitude for all j, k and (ii) the correlation between $f(X_n)$ and the sample mean in (16.2) goes to zero as $n \to \infty$ (Parzen, 1962, pp. 72–75). In other words, the process is ergodic in the m.s. sense if there is less correlation between the terminal observation $f(X_n)$ and the sample mean in (16.2) as n is increased. In the MCMC methods, if the Markov chain is generated properly, then $p^*(\cdot)$ equals the target density $p(\cdot)$, as desired. That is, the limit of the ergodic mean in (16.2) corresponds to the desired value $E[f(X)]$ computed with respect to $p(\cdot)$.

We outlined above a general formulation for generating random samples via the output of a Markov chain. There are, of course, some fundamental elements that need to be specified to justify the approach and to provide the details required for implementation. These elements form the basis for the MCMC approach.

In the remainder of this chapter devoted to MCMC, we present an overview of the Metropolis–Hastings (M-H) and Gibbs sampling implementations of MCMC, discuss applications in Bayesian modeling, sketch the theoretical foundations, and present some examples. There are also some important aspects of MCMC that we do not cover in this relatively brief review. For example, we do not discuss formal methods for diagnosing convergence. This is an active area of current research, with a large number of specialized techniques. Robert and Casella (1999, Chap. 8) is one of many references on this subject. As with general stochastic search and optimization methods, there is no universal method for knowing when to stop a chain. We also do not review the many software packages that are available for both simple and sophisticated implementations of MCMC.[2]

16.2 METROPOLIS-HASTINGS ALGORITHM

As discussed above, the sum of dependent random variables in (16.2) converges to the mean of $f(X)$ under appropriate conditions. Although this is hopeful, we must show that this is the "right" mean, that is, the mean computed with respect to $p(x)$. Fortunately, it is surprisingly easy to produce such a Markov process via a variant of the Metropolis sampling seen in the simulated annealing algorithm of Chapter 8. The form given here, introduced by Hastings (1970), builds on the criterion in Metropolis et al. (1953).

Given an initial condition X_0, the M-H algorithm is a mechanism for producing the Markov process $X_1, X_2, \ldots$ for use in (16.2). From a state $X_k = x$, the next state is chosen by generating a candidate point W from a *proposal distribution* (sometimes called an *instrumental distribution* or a *candidate-generating distribution*), $q(\cdot|x)$. In principle, the proposal distribution may be chosen arbitrarily, although there may be efficiency advantages to one form over another in some applications. The proposal distribution satisfies the key property for density functions, namely

$$\int q(w \mid X = x) dw$$

for any $X = x$, as appropriate. There are also some very modest conditions for the proposal distribution, as discussed in Robert and Casella (1999, pp. 233–235). For example, the set of points w where $q(w|x) > 0$, as $X = x$ ranges over the set of points where $f(x) \neq \mathbf{0}$, should be a *superset* of the set of points x where $f(x) \neq \mathbf{0}$. A common example for $q(\cdot|x)$ is a uniform distribution centered around x. This example satisfies the superset condition. One implication of the superset and other conditions is that it is possible to generate candidate points that "fill up" the support of the target density (i.e., provide an adequate number of points throughout the region where $p(\cdot) > 0$).

[2] Let us mention one prominent package, however. BUGS (*B*ayesian inference *U*sing *G*ibbs *S*ampling) is available free on the Web and is one of the most popular standard packages. BUGS is available at *www.mrc-bsu.cam.ac.uk/bugs*.

Analogous to step 2 of the simulated annealing algorithm in Section 8.2, the candidate point $\boldsymbol{W}$ is accepted with probability $\rho(\boldsymbol{X}_k, \boldsymbol{W})$, where

$$\rho(\boldsymbol{x}, \boldsymbol{w}) = \min\left\{\frac{p(\boldsymbol{w})}{p(\boldsymbol{x})}\frac{q(\boldsymbol{x}\,|\,\boldsymbol{w})}{q(\boldsymbol{w}\,|\,\boldsymbol{x})},\ 1\right\}. \qquad (16.3)$$

If the candidate point is accepted, then $\boldsymbol{X}_{k+1} = \boldsymbol{W}$; otherwise, $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$. Note that $q(\cdot\,|\,\cdot)$, appearing in both the numerator and denominator of (16.3), is the *same function* with only the conditioning interchanged ($\boldsymbol{x}$ and $\boldsymbol{w}$ have the same dimension). (In general, of course, the functional *form* for the distribution of one random variable conditioned on another depends on the order of the conditioning.) Likewise, $p(\cdot)$ in the numerator and denominator is the same function with only the arguments changed. (The connection to simulated annealing is discussed further in Robert and Casella, 1999, p. 281.)

Because of the ratio form, an important implication of (16.3) is that one only needs to know $p(\cdot)$ to within a constant because the constant cancels out. Eliminating the need to determine the constant has significant practical advantages by removing the need for a formidable numerical integration. For example, in Bayesian applications, as considered in Section 16.6, the target density represents a posterior density that is conditioned on some set of data (the data conditioning does not affect the mechanics of the M-H algorithm). It is notoriously difficult to obtain the constant associated with the posterior density function because the constant is the marginal density for the data appearing in the denominator of Bayes' rule. This marginal density requires difficult numerical integration. We discuss this further in Section 16.6.

Table 16.1 summarizes two common proposal distributions. Suppose that $m = \dim(\boldsymbol{X})$ (so $m = \dim(\boldsymbol{W})$). Let us denote an m-fold uniform distribution by $U_m(\boldsymbol{a}, \boldsymbol{b})$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are m-dimensional vectors. This distribution is such that each component of the m-dimensional random vector has an independent uniform distribution with lower and upper endpoints given by the corresponding component of $\boldsymbol{a}$ and $\boldsymbol{b}$ (so the probability is uniform over the hypercube defined

Table 16.1. Examples of two popular general forms for proposal distributions.

General form of proposal distribution	$q(\boldsymbol{w}\|\boldsymbol{x})$	$q(\boldsymbol{x}\|\boldsymbol{w})$
Normal with covariance matrix $\boldsymbol{\Sigma}$	$N(\boldsymbol{x}, \boldsymbol{\Sigma})$	$N(\boldsymbol{w}, \boldsymbol{\Sigma})$
Uniform of width 2δ for each component	$U_m(\boldsymbol{x}-\delta\boldsymbol{1}_m,\ \boldsymbol{x}+\delta\boldsymbol{1}_m)$	$U_m(\boldsymbol{w}-\delta\boldsymbol{1}_m,\ \boldsymbol{w}+\delta\boldsymbol{1}_m)$

by $\boldsymbol{a}$ and $\boldsymbol{b}$). These two candidate-generating processes are examples of *random walk processes*. That is, the candidate point may be written as the current point plus noise: $\boldsymbol{W} = \boldsymbol{X}$ + noise, where the noise is a mean-zero normal or uniform distribution. Note further that $q(\boldsymbol{w}|\boldsymbol{x}) = q(\boldsymbol{x}|\boldsymbol{w})$, an example of the important special case where the proposal distribution is symmetric. An implication of the symmetric proposal is that criterion (16.3) simplifies to

$$\rho(\boldsymbol{x}, \boldsymbol{w}) = \min\left\{\frac{p(\boldsymbol{w})}{p(\boldsymbol{x})},\ 1\right\}$$

as a result of $q(\boldsymbol{x}\,|\,\boldsymbol{w})/q(\boldsymbol{w}\,|\,\boldsymbol{x}) = 1$. For the m-fold uniform distribution in Table 16.1, let $\mathbf{1}_m$ denote an m-dimensional vector of 1's and δ be a positive constant.

A remarkable result associated with (16.3) is that although the proposal distribution $q(\cdot\,|\,\cdot)$ may have almost any form (subject to the modest conditions mentioned above), the stationary distribution of the chain satisfies $p^*(\cdot) = p(\cdot)$ (Hastings, 1970). Chib and Greenberg (1995) and Robert and Casella (1999, pp. 235–238) elaborate on some of the arguments in Hastings (1970), establishing an appropriate form of convergence in distribution. Below is a summary of the steps for the M-H algorithm. The first several steps pertain to the burn-in period and the remaining steps are used to form the ergodic average in (16.2).

M-H Algorithm for Estimating $E[f(\boldsymbol{X})]$

Step 0 **(Initialization)** Choose the length of the burn-in period M and an initial state $\boldsymbol{X}_0$. Set $k = 0$.

Step 1 Generate a candidate point $\boldsymbol{W}$ according to the proposal distribution $q(\cdot\,|\boldsymbol{X}_k)$.

Step 2 Generate a point U from a $U(0, 1)$ distribution. Set $\boldsymbol{X}_{k+1} = \boldsymbol{W}$ if $U \leq \rho(\boldsymbol{X}_k, \boldsymbol{W})$ from (16.3). Otherwise, set $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$.

Step 3 Repeat steps 1 and 2 until $\boldsymbol{X}_M$ is available. Terminate the burn-in process and proceed to step 4 with $\boldsymbol{X}_k = \boldsymbol{X}_M$.

Step 4 Carry out step 1.

Step 5 Carry out step 2.

Step 6 Repeat steps 4 and 5 until it is possible to compute the ergodic average of $n - M$ evaluations in (16.2). (Of course, if desired, this average can be computed recursively without storing all of $f(\boldsymbol{X}_{M+1}), f(\boldsymbol{X}_{M+2}),\ldots, f(\boldsymbol{X}_n)$.) This ergodic average is the estimate of $E[f(\boldsymbol{X})]$ under the target density $p(\cdot)$.

There are a number of specific ways in which the overall M-H algorithm can be implemented. The most obvious variation in implementation is in the choice of the proposal distribution $q(\cdot\,|\,\cdot)$. Although almost any choice of $q(\cdot\,|\,\cdot)$ works in the sense that the ergodic average in (16.2) converges to $E[f(\boldsymbol{X})]$, there are clear differences in the rate of convergence depending on the nature of the problem. There are also forms of averaging that differ from the standard ergodic averaging. One variation is to run many independent chains, with each chain terminating at X_{M+1}. In this way, $E[f(X)]$ is estimated by forming a sample mean of *independent* values $f(X_{M+1})$. Regeneration, as discussed in Section 14.2, may also be used to improve the performance of M-H (Gilks et al., 1998). This creates independent blocks of iterations, allowing for the proposal distribution to be adapted at each block to improve the sampling.

The rate at which candidate values W are accepted affects M-H performance. This rate should be neither too small nor too large to allow for an adequate exploration of the space. As summarized in Roberts et al. (1997) and Roberts and Rosenthal (1998), if the proposal distribution is normal, the approximate optimal acceptance rate is 23 percent as $\dim(X) \to \infty$ (i.e., 23 percent of the time, $X_{k+1} \neq X_k$ in the M-H steps). In addition, if the target distribution is also normal, the approximate optimal acceptance rate is 45 percent when $\dim(X) = 1$ (Chen et al., 2000, p. 23). These results pertain to the *random walk* proposal distribution, implying that the target and proposal distributions are not identical to each other (if they were identical, a 100 percent acceptance rate is optimal!). Roberts and Rosenthal (2001) point out that the 23 percent rate is, in fact, quite robust to deviations from the assumptions above (e.g., the optimal acceptance rate for $\dim(X) = 10$ is negligibly different from the asymptotic rate of 23 percent; further, the results apply for certain types of nonnormal proposal distributions). This robustness can be seen both theoretically and empirically. As we have seen with other aspects of stochastic search and optimization, tuning is necessary to achieve approximately optimal performance. In practice, this can be achieved by running test iterations of the algorithm, adjusting parameters in the proposal distribution until the acceptance rate is near the optimal rate.

Let us present a simple example where the target density is bivariate normal. This example demonstrates the performance of M-H in a setting that is easy to understand. In practice, there are other (likely more) efficient methods of generating samples from a multivariate normal distribution (see Appendix D).

Example 16.1—Simulating a bivariate normal distribution. Consider a problem where the target density is bivariate normal with the two variables highly correlated. In particular, suppose that

$$X \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.9\\0.9 & 1\end{bmatrix}\right).$$

Further, suppose that the proposal distribution $q(w|x)$ is a shifted uniform distribution as in Table 16.1. That is, W (conditioned on $X = x$) is generated according to $U_2(x - 0.5\mathbf{1}_2, x + 0.5\mathbf{1}_2)$. The distribution for each of the two elements in W has the standard unit length (centered around the elements of the moving point x). Because $q(w|x) = q(x|w)$, the form of the normal target density function implies that

$$\rho(\boldsymbol{x},\boldsymbol{w}) = \min\left\{\frac{\exp\left(-\frac{1}{2}\boldsymbol{w}^T\begin{bmatrix}1 & 0.9\\ 0.9 & 1\end{bmatrix}^{-1}\boldsymbol{w}\right)}{\exp\left(-\frac{1}{2}\boldsymbol{x}^T\begin{bmatrix}1 & 0.9\\ 0.9 & 1\end{bmatrix}^{-1}\boldsymbol{x}\right)},\ 1\right\}.$$

As discussed above, note that the constant terms in the bivariate normal density functions are not needed in constructing $\rho(\cdot)$ (a trivial advantage here, but a major advantage in applications such as Bayesian analysis).

Figure 16.1 shows the results of a study where M-H was used to estimate $E[f(X)]$, where $f(X) = [1, 1]X$. Thus, we are estimating the sum of the means for the two elements of X. The figure shows the evolution of the ergodic averages for three independent runs. The $M = 500$ burn-in period for each run is initialized at $X_0 = [-1, 1]^T$. It is clear that the three runs are all settling down near the true value $E[f(X)] = 0$. Nevertheless, some improvement in the estimates is possible, especially for the run represented by the light-gray line.

Tuning of the proposal distribution $U_2(\cdot)$ from the current naïve unit-length intervals provides better results. For each of the runs, about 70 percent of the candidate points W are accepted. This rate is higher than the above-mentioned optimal rate of 23 percent under a normal proposal and large dim(X)

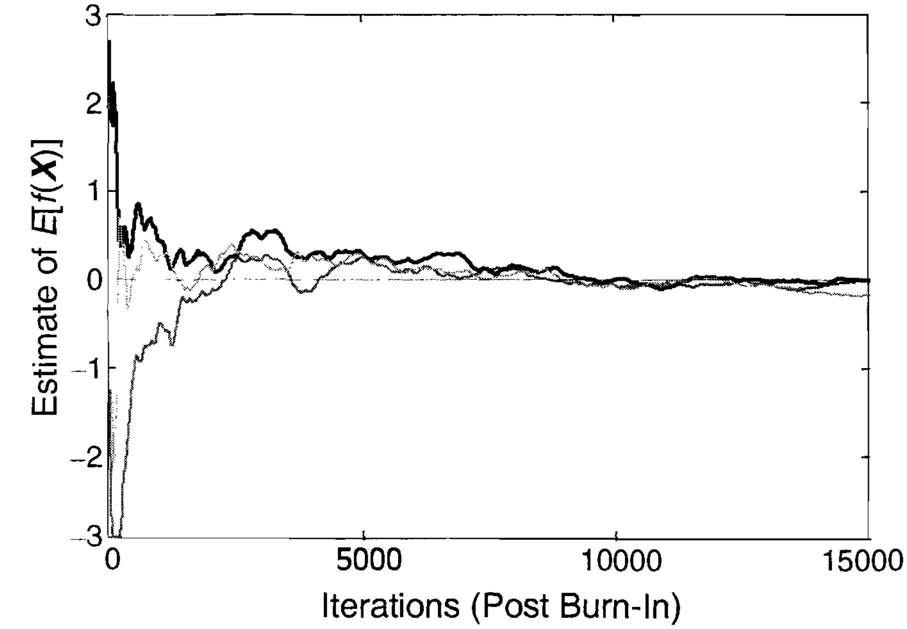

Figure 16.1. Traces for three independent runs of M-H sampler in estimating $E[f(X)]$ (i.e., estimating the sum of the two components in $E(X)$). Target value is 0. Burn-in period (M) is 500 for each run. Standard (unit length) uniform proposal distribution used here. Improved performance is possible by optimizing the proposal distribution.

(neither of which are true here). Based on numerical experimentation, a wider support for the uniform distribution decreases the acceptance rate in this case. In fact, as a testament to the robustness of the 23 percent guideline (see the discussion above), a decreased acceptance rate *did* improve the performance in this case even though the proposal distribution is not normal and dim(X) is not large. Changing the proposal distribution to $U_2(\boldsymbol{x} - 2\mathbf{1}_2, \boldsymbol{x} + 2\mathbf{1}_2)$ caused the acceptance rate to drop to approximately 24 percent and decreased the magnitude of the maximum terminal error in 50 independent runs of M-H from 0.33 to 0.20. Other *forms* for the proposal distribution may also be valuable; as mentioned earlier, the normal distribution is a frequent proposal form. (Exercise 16.2 considers the normal proposal form and the wider $U_2(\boldsymbol{x} - 2\mathbf{1}_2, \boldsymbol{x} + 2\mathbf{1}_2)$ form.)

While the M-H algorithm has wide applicability, there is some loss of information in the ergodic sampling (with its serial dependence) relative to independent sampling. For comparison purposes, suppose that one could directly obtain independent samples of $\boldsymbol{X}$ (an idealized case since one would then not need M-H!). An estimate of $E[f(X)]$ based on the mean of 15,000 independent samples of $[1, 1]\boldsymbol{X}$ would have a standard deviation of 0.0159 (Exercise 16.1). This contrasts with an approximate standard deviation of 0.138 for the terminal estimate from 15,000 samples from the M-H algorithm as represented in Figure 16.1. This approximation is the sample standard deviation of the terminal estimate from 50 independent runs of M-H. Hence, for the same number of samples (ignoring the burn-in samples), the M-H algorithm produces an estimate over eight times more variable than direct sampling ($0.138/0.0159 \approx 8.7$). The increased variation in the M-H estimate is due to the positive correlation in the iterates (Exercise 16.1). Of course, M-H is predicated on direct sampling not being available, so this type of comparison is only useful to establish a bound on behavior. ❑

We next consider what is likely the most popular specific implementation of the M-H algorithm.

16.3 GIBBS SAMPLING

Gibbs sampling represents an implementation of the M-H algorithm on an element-by-element basis for the components in $\boldsymbol{X}$. The term *Gibbs sampling* was introduced by Geman and Geman (1984) in a specific implementation of a Gibbs distribution for sampling on lattices.[3] The term is now used more generally (and casually) to refer to the special case where the proposal distribution is built directly from the density of interest $p(\cdot)$. Because of this restriction, the method is more limited than M-H. The restriction sometimes leads to advantages in efficiency and ease of implementation via the elimination of the tuning typically needed in M-H. Gibbs sampling is especially important in

[3] Gibbs sampling derives its name from the physicist Josiah W. Gibbs, 1839–1903, based on the connection to Gibbs random fields identified in Geman and Geman (1984).

Bayesian implementations. Gibbs samplinp is uniquely designed for multivariate problems. In fact, "...the crucial issue is ıeplacement of the sampling of a high-dimensional vector with sampling of lower-dimensional component blocks, thus breaking the so-called curse of dimensionality" (Gelfand, 2000).

Gibbs sampling may be interpreted as a concatenation of m M-H algorithms, one for each variable in the random vector of interest, $\boldsymbol{X}$. (Because this version of M-H does not generate a new vector in toto, it is not precisely the same as the multivariate form in Section 16.2.) This concatenation has m target distributions, each representing a conditional distribution for one variable given values for all other variables (called the *full conditional distribution*, as discussed below). In contrast to the M-H algorithm, where the proposal distributions may be chosen almost arbitrarily, the proposals here have a required form. Namely, the proposal distribution for the ith element of $\boldsymbol{X}$ is the conditional distribution of that variable given the most recent values for all other variables during the iterative process. Thus, the target and proposal distributions are the same.

To make the above concepts more concrete, consider a trivariate ($m = 3$) problem based on density functions, the three variables being X, Y, and Z. The three target densities are $p_{X|Y,Z}(\cdot\,|\,\cdot)$, $p_{Y|X,Z}(\cdot\,|\,\cdot)$, and $p_{Z|X,Y}(\cdot\,|\,\cdot)$. As with generic M-H, let W represent a candidate random variable generated according to the candidate density. For the first element, X, the candidate-generating density is then

$$q(w\,|\,y,z) = p_{X|Y,Z}(w\,|\,y,z). \tag{16.4}$$

With the target density $p(\cdot) = p_{X|Y,Z}(\cdot\,|\,\cdot)$, the substitution of (16.4) into the probability of acceptance for the M-H algorithm as given in (16.3) yields

$$\rho(x,w) = \min\left\{\frac{p_{X|Y,Z}(w\,|\,y,z)}{p_{X|Y,Z}(x\,|\,y,z)}\frac{q(x\,|\,y,z)}{q(w\,|\,y,z)},\ 1\right\} = \min\{1,\ 1\} = 1. \tag{16.5}$$

Unlike the general M-H algorithm, relationship (16.5) implies that the new point W is always accepted as a representation for X. Hence, from (16.5), given any previous values for Y and Z, the candidate value W generated from $q(w\,|\,y,z) = p_{X|Y,Z}(w\,|\,y,z)$ is guaranteed to be the new value for X. Identical arguments apply for the other two variables, Y and Z. In fact, because of the automatic acceptance of a candidate point, there is no need for a separate (W) candidate process for each variable; one simply generates a new X, Y, or Z as appropriate. The general multivariate relationship between Gibbs sampling and M-H is discussed in Robert and Casella (1999, pp. 296–297) and Gilks et al. (1996, pp. 10–12). This relationship is an obvious extension of the trivariate setting above.

As a consequence of the theory of Markov processes, the iterative values X_k, Y_k, Z_k generated via the Gibbs sampler represent, in the limit, observations from the joint density $p_{X,Y,Z}(x,y,z)$. Further, each of X_k, Y_k, and Z_k has a distribution that approaches its respective marginal, $p_X(x)$, $p_Y(y)$, and $p_Z(z)$.

The implementation of the Gibbs sampler for the trivariate problem of generating samples from $p_{X,Y,Z}(\cdot)$ is as follows. Suppose that the sampler begins with an initial guess at Y and Z, say Y_0 and Z_0. Using the full conditional $p_{X|Y,Z}(x|Y_0, Z_0)$, we can then generate a sample point X_1 by Monte Carlo. We next use the full conditional $p_{Y|X,Z}(y|X_1, Z_0)$ to generate a sample point Y_1. Likewise, $p_{Z|X,Y}(z|X_1, Y_1)$ is used to generate Z_1. At this point, the Gibbs sampler has completed one iteration, producing X_1, Y_1, and Z_1. We now repeat the process, using Y_1 and Z_1 to initiate the conditioning and producing a sample X_2, Y_2, and Z_2. This Monte Carlo sampling forms a sequence

$$Y_0, Z_0; X_1, Y_1, Z_1; X_2, Y_2, Z_2; \ldots; X_n, Y_n, Z_n. \tag{16.6}$$

The sequence in (16.6) is called the *Gibbs sequence*. The sequential sampling approach above extends in an obvious way to the general m-dimensional case.

The Gibbs sequence above can be used in several ways to estimate $E[f(X, Y, Z)]$. For example, let M denote the burn-in period, as with the M-H algorithm. Under modest conditions, the later observations, X_{M+1}, Y_{M+1}, $Z_{M+1}; \ldots;$ X_n, Y_n, Z_n, in the Gibbs sequence represent measurements from $p_{X,Y,Z}(\cdot)$. These can then be substituted in the averaging of (16.2) to produce an estimate of $E[f(X, Y, Z)]$. A variation on standard ergodic averaging is to pick off every (say) ℓth value in the chain, and average only these values. If ℓ is reasonably large, this is roughly equivalent to averaging independent samples. Yet another way of estimating $E[f(X, Y, Z)]$ is to generate N independent Gibbs sequences, using only the final output in each sequence. Now, $E[f(X, Y, Z)]$ is estimated by an average of N samples, where the summands are i.i.d.

The example below shows the derivation of the required conditional distributions for generating the Gibbs sequence in a bivariate ($m = 2$) setting. This example pertains to a problem with discrete outcomes.

Example 16.2—Gibbs sampling with a Bernoulli distribution. The sampling need not be performed according to continuous random variables and associated probability *density* functions. Consider the following example associated with a Bernoulli distribution, as given in Casella and George (1992). Let the bivariate random vector $[X, Y]^T$ have a joint probability (mass) function

$$\begin{bmatrix} P(X=0, Y=0) & P(X=0, Y=1) \\ P(X=1, Y=0) & P(X=1, Y=1) \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix},$$

where $p_{ij} \geq 0$ and $p_{00} + p_{01} + p_{10} + p_{11} = 1$. That is, each variable, X or Y, is a Bernoulli-distributed random variable that can take on a value 0 or 1. The two variables are dependent. So, for example, the likelihood of $Y = 1$ depends on whether $X = 0$ or 1.

Using basic rules of probability, it is simple to compute the *conditional* probabilities $P(Y = a|X = b)$ and $P(X = a|Y = b)$, where a and b are 0 or 1. These probabilities are given in the two matrices below:

$$\begin{bmatrix} P(Y=0|X=0) & P(Y=1|X=0) \\ P(Y=0|X=1) & P(Y=1|X=1) \end{bmatrix} = \begin{bmatrix} \dfrac{p_{00}}{p_{00}+p_{01}} & \dfrac{p_{01}}{p_{00}+p_{01}} \\ \dfrac{p_{10}}{p_{10}+p_{11}} & \dfrac{p_{11}}{p_{10}+p_{11}} \end{bmatrix}$$

and

$$\begin{bmatrix} P(X=0|Y=0) & P(X=1|Y=0) \\ P(X=0|Y=1) & P(X=1|Y=1) \end{bmatrix} = \begin{bmatrix} \dfrac{p_{00}}{p_{00}+p_{10}} & \dfrac{p_{10}}{p_{00}+p_{10}} \\ \dfrac{p_{01}}{p_{01}+p_{11}} & \dfrac{p_{11}}{p_{01}+p_{11}} \end{bmatrix}.$$

The Gibbs sampling procedure applied with these two sets of conditional probabilities yields a sequence of 0's and 1's. We can then use this sequence to estimate $E[f(X, Y)]$. ❑

The above two- and three-variable discussions serve to motivate the general multivariate setting. In this more general setting, we continue to be interested in estimating quantities of the form $E[f(\boldsymbol{X})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$ (or its discrete analogue), where $\boldsymbol{X}$ is a collection of m (say) univariate or multivariate components. In the common case where the m components are univariate, the kth sample $\boldsymbol{X}$ from the Gibbs sampling algorithm is:

$$\boldsymbol{X}_k = \begin{bmatrix} X_{k1} \\ X_{k2} \\ \vdots \\ X_{km} \end{bmatrix},$$

where X_{ki} denotes the ith component for the kth replicate of $\boldsymbol{X}$ generated via the sampling algorithm. As we saw above, in Gibbs sampling, it is not necessary to introduce a separate candidate process $\boldsymbol{W}$ (as in the general M-H approach) because the candidate point is always accepted. While each X_{ki} is usually a scalar element, there are problems where it is useful to have at least some X_{ki} be multivariate (see Example 16.7).

The idea of the proposal distribution for M-H can be used to update each component of $\boldsymbol{X}$. This updating is done on a component-by-component basis. Central to the updating is the conditional random variable $\{X_{k+1,i}|\boldsymbol{X}_{k\backslash i}\}$, where

$$\boldsymbol{X}_{k\backslash i} \equiv \{X_{k+1,1}, X_{k+1,2}, \ldots, X_{k+1,i-1}, X_{k,i+1}, \ldots, X_{km}\},$$

$i = 1, 2, \ldots, m$. To avoid very cumbersome subscript notation associated with the relevant random variables and the associated conditioning, let $p_i(\cdot)$ represent the sampling density for the conditional random variable:

$$\{X_{k+1,i} | \boldsymbol{X}_{k\backslash i}\} \sim p_i(x | \boldsymbol{X}_{k\backslash i}),\ i = 1, 2, \ldots, m.$$

That is, $p_i(\cdot)$ represents the sampling density for the random variable $X_{k+1,i}$ conditioned on $\boldsymbol{X}_{k\backslash i}$, where the first $i-1$ elements of $\boldsymbol{X}_{k\backslash i}$ represent sample points at the same $(k+1)$st iteration while the remaining $m-i$ elements are points available from the kth iteration. This strange-looking conditioning follows naturally from the sequential component-wise processing in the Gibbs sampling procedure, as given below. The conditioning represents the most recent information available when generating the ith component of $\boldsymbol{X}$.

The density $p_i(\cdot)$ is the generalization of the full conditional densities that were given above for the $m = 3$ problem. Note that the full conditional is a *univariate* sampling density in the common case where each X_{ki} is univariate. Thus, even when $\boldsymbol{X}$ is high dimensional, the sampling is univariate. This has significant potential advantages. The derivation of the full conditional follows from basic laws of probability:

$$p_i(x | \boldsymbol{X}_{k\backslash i}) = \frac{p_{\boldsymbol{X}}(x, \boldsymbol{X}_{k\backslash i})}{\int p_{\boldsymbol{X}}(x, \boldsymbol{X}_{k\backslash i})\, dx}, \tag{16.7}$$

where the denominator integral is over the domain for $X_{k+1,i} = x$, and $p_{\boldsymbol{X}}(\cdot)$ appearing on the right-hand side represents (as before) the density for $\boldsymbol{X}$. In some practical applications, this definition can be used directly to obtain the full conditionals required for the steps of the Gibbs sampling procedure. In other cases, standard numerical methods for random number generation can be used. A good discussion of methods for obtaining samples from full conditionals appears in Gilks et al. (1996, Chap. 5). There are some applications for which sampling from full conditionals is difficult. One popular method for coping with such a difficulty is the *Metropolis within Gibbs* approach. This method involves the use of M-H within the Gibbs sampling steps below to produce samples from the full conditionals (see, e.g., Gilks et al., 1996, pp. 84–85; Robert and Casella, 1999, pp. 322–326).

Let us now present a standard implementation of the Gibbs sampling algorithm for estimating $E[f(\boldsymbol{X})]$ in (16.1). After presenting these steps, we comment on several variations of the standard algorithm.

Gibbs Sampling Algorithm for Estimating $E[f(X)]$

Step 0 (**Initialization**) Choose the length of the burn-in period M and an arbitrary initial state $\boldsymbol{X}_0$. Set $k = 0$.

Step 1 Generate $\boldsymbol{X}_{k+1}$ according to the following m steps:
1. Generate $X_{k+1,1} \sim p_1(x \mid \boldsymbol{X}_{k\backslash 1})$.
2. Generate $X_{k+1,2} \sim p_2(x \mid \boldsymbol{X}_{k\backslash 2})$.
$\vdots$
***m*.** Generate $X_{k+1,m} \sim p_m(x \mid \boldsymbol{X}_{k\backslash m})$.

Step 2 Repeat step 1 until $\boldsymbol{X}_M$ is available. Terminate the burn-in process and proceed to step 3 with $\boldsymbol{X}_k = \boldsymbol{X}_M$.

Step 3 Carry out step 1.

Step 4 Repeat step 3 until it is possible to compute the ergodic average of $n - M$ evaluations in (16.2). (Of course, if desired, this average can be computed recursively without storing all of $f(\boldsymbol{X}_{M+1}), f(\boldsymbol{X}_{M+2}), \ldots, f(\boldsymbol{X}_n)$.) This ergodic average is the estimate of $E[f(X)]$ under the target distribution $p(\cdot)$.

As with M-H, there are implementations of Gibbs sampling for estimating $E[f(X)]$ other than the standard ergodic averaging of (16.2). One variation is to run many independent chains, each chain terminating at $\boldsymbol{X}_{M+1}$. In this way, $E[f(X)]$ is estimated by forming a sample mean of *independent* values $f(\boldsymbol{X}_{M+1})$. Another variation is to average two dependent chains, where the dependence is introduced via the notion of antithetic random variables (an analogue of common random numbers discussed in Chapter 14). Such antithetic averaging based on pairs of dependent chains can lead to significant reduction of the variance of the estimate for $E[f(X)]$ (Frigessi et al., 2000).

16.4 SKETCH OF THEORETICAL FOUNDATION FOR GIBBS SAMPLING

It is not immediately obvious why the above Markov sampling procedure works. How can sequential sampling from *conditional* distributions produce samples from a *joint* distribution? Let us sketch the rationale. Given the close connection of Gibbs sampling to the M-H algorithm introduced earlier, the arguments here also provide some flavor of the basis for M-H, although the details are somewhat different. This relatively informal discussion is a simplified version of the discussion in Gelfand and Smith (1990) and Robert and Casella (1999, Sect. 7.1.3).

Let us consider the three-variable case, X, Y, and Z, and sketch how the sampling from the full conditionals as above provides the information necessary to obtain samples from the density $p_{X,Y,Z}(x,y,z)$ (or from the marginals for any of the three variables). The ideas for three variables below extend immediately to an arbitrary number of variables (m as above). From basic rules of conditional probability,

$$p_X(x) = \int p_{X|Y,Z}(x\,|\,y,z)p_{Y,Z}(y,z)\,dy\,dz\,,$$

$$p_Y(y) = \int p_{Y|X,Z}(y\,|\,x,z)p_{X,Z}(x,z)\,dx\,dz\,,$$

$$p_Z(z) = \int p_{Z|X,Y}(z\,|\,x,y)p_{X,Y}(x,y)\,dx\,dy\,,$$

where the integrals are over the relevant domains in $\mathbb{R}^2$. Note the presence of a full conditional in each of the integrands above. The full conditionals form the basis for the Markov aspect of the sampling because the next random variate is generated based on only the most recent conditioning.

The expressions above provide the basis for the Gibbs sampling in (16.6). Suppose that we begin the sampler with the top expression above and make an initial guess at Y and Z, say Y_0 and Z_0. Using the full conditional $p_{X|Y,Z}(x|Y_0,Z_0)$, we can then generate a sample point X_1 by Monte Carlo. Proceeding downward through the expressions above, we next use the full conditionals $p_{Y|X,Z}(y|X_1,Z_0)$ and $p_{Z|X,Y}(z|X_1,Y_1)$ to generate sample points Y_1 and Z_1, respectively. At this point, we have completed one iteration of the Gibbs sampler, producing a sample X_1, Y_1, and Z_1. The process is then repeated, using Y_1 and Z_1 to initiate the conditioning and producing a sample X_2, Y_2, and Z_2. Continuing this process, it can be shown under modest conditions that X_k, Y_k, and Z_k jointly converge in distribution as $k \to \infty$ to the distribution associated with $p_{X,Y,Z}(x,y,z)$. Likewise, X_k, Y_k, and Z_k individually converge in distribution to the respective marginal distributions associated with $p_X(x)$, $p_Y(y)$, and $p_Z(z)$ (see Robert and Casella, 1999, Sect. 7.1.3).

The formal basis for convergence follows from the *fixed-point integral equation* that establishes a relationship between marginal and conditional distributions. For instance, if the marginal $p_X(x)$ in the three-variable problem above is of interest, then the Gibbs sampling routine provides a sample having limiting density $p_X(x)$, where the *function* $p_X(\cdot)$ is the unique solution (i.e., $\phi(\cdot) = p_X(\cdot)$) to the integral equation:

$$\begin{aligned}\phi(\cdot) &= \int\Big[\int p_{X|Y,Z}(\cdot\,|\,y,z)p_{Y,Z|X}(y,z\,|\,\tau)\,dy\,dz\Big]\phi(\tau)\,d\tau \\ &\equiv \int K(\cdot\,|\,\tau)\phi(\tau)\,d\tau,\end{aligned}$$

and where the integral inside the [ ] in the top expression, represented by $K(\cdot\,|\tau)$, is over the appropriate subspace of $\mathbb{R}^2$ (Tanner and Wong, 1987; Gelfand and Smith, 1990). The term $K(\cdot)$ is often called the *transition kernel*. From the expression above, it can be shown (Tanner and Wong, 1987; Gelfand and Smith, 1990) that the Gibbs recursion at a specific x can be written in Markov transition form as

$$\phi_{k+1}(x) = \int K(x\,|\,\tau)\phi_k(\tau)d\tau,$$

where $\phi_k(\cdot)$ denotes the true density for X_k. The fundamental result in Gibbs sampling is that $\phi_k(\cdot)$ converges to $p_X(\cdot)$ as $k \to \infty$. Similar ideas apply—with analogous transition kernels—for other target densities and dimensions $m \neq 3$.

Analogous situations apply when the random variables have a discrete distribution. Here the kernel-based form above is replaced with a Markov transition matrix (Appendix E). Consider, for example, the scalar element X in the trivariate illustration above. Let $\boldsymbol{p}_k$ represent the vector of probabilities associated with the possible outcomes for X_k (so, e.g., if X is a random variable having 10 possible outcomes, there are 10 nonnegative elements in $\boldsymbol{p}_k$ for all k, with the elements summing to 1). Let $\boldsymbol{P}$ represent the transition matrix governing the probability of going from X_k to X_{k+1} (so $\boldsymbol{P}$ has dimension $\dim(\boldsymbol{p}_k)\times\dim(\boldsymbol{p}_k)$). The elements of this transition matrix are directly available through the conditional probabilities of the variables in the problem. Thus, in the trivariate setting above, an individual element of $\boldsymbol{P}$ is available by applying the total probability theorem to first determine the probabilities of going from X_k to Y_{k+1}, then from Y_{k+1} to Z_{k+1}, and finally, from Z_{k+1} to X_{k+1}.

By standard Markov chain theory,

$$\boldsymbol{p}_{k+1}^T = \boldsymbol{p}_k^T\boldsymbol{P} = \boldsymbol{p}_0^T\boldsymbol{P}^{k+1}.$$

Then, if all elements of $\boldsymbol{P}$ are strictly positive, $\boldsymbol{p}_k$ converges to the limiting $\bar{\boldsymbol{p}}$ that is the solution to the balance equation

$$\bar{\boldsymbol{p}}^T = \bar{\boldsymbol{p}}^T\boldsymbol{P} \tag{16.8}$$

(Theorem E.1 in Appendix E). In particular, the $\bar{\boldsymbol{p}}$ that satisfies this stationarity condition must be the marginal distribution for X. Thus, the Gibbs sampler converges to the marginal distribution, as desired. Building on Example 16.2, let us now illustrate this balance equation.

Example 16.3—Balance equation for discrete bivariate problem. In the context of Example 16.2, suppose that the matrix of joint probabilities is

$$\begin{bmatrix} P(X=0,Y=0) & P(X=0,Y=1) \\ P(X=1,Y=0) & P(X=1,Y=1) \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.5 & 0.2 \end{bmatrix}.$$

The vector of marginal probabilities associated with X is

$$\begin{bmatrix} P(X=0) \\ P(X=1) \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}.$$

We are interested in the 2×2 transition matrix $\boldsymbol{P}$ governing the probability of going from X_k to X_{k+1}. From basic laws of conditional probability, the $(1,1)$ component of $\boldsymbol{P}$ is given by

$$\begin{aligned} P(X_{k+1}=0 \mid X_k=0) = P(X_{k+1}=0 \mid Y_{k+1}=0)P(Y_{k+1}=0 \mid X_k=0) \\ + P(X_{k+1}=0 \mid Y_{k+1}=1)P(Y_{k+1}=1 \mid X_k=0). \end{aligned}$$

The other three components of $\boldsymbol{P}$ are found in a like manner. The X to Y and Y to X transition matrices in Example 16.2 provide the required conditional probabilities (Exercise 16.5). It is found that

$$\boldsymbol{P} = \begin{bmatrix} 0.3889 & 0.6111 \\ 0.2619 & 0.7381 \end{bmatrix}.$$

One may verify that $\bar{\boldsymbol{p}}^T = \bar{\boldsymbol{p}}^T\boldsymbol{P}$, where $\bar{\boldsymbol{p}}^T = [P(X=0), P(X=1)] = [0.3, 0.7]$, indicating convergence of the Gibbs sampler. ❑

16.5 SOME EXAMPLES OF GIBBS SAMPLING

This section presents three examples of Gibbs sampling. The first is for a (multivariate) normal target distribution, which leads to conditional distributions that are also normal. The second is for a truncated exponential distribution. One point illustrated in the second example is that the target $p(\boldsymbol{x}) = p_{\boldsymbol{X}}(\boldsymbol{x})$ does not automatically exist even when the full conditionals do exist. The Gibbs sampler only produces meaningful results (of course!) if the target distribution exists. General regularity conditions for the existence of the target distribution (which is often the joint density for the elements in $\boldsymbol{X}$) are beyond the scope of the treatment here, but may, for example, be found in Robert and Casella (1999, Sect. 7.1.5). In the special case of two variables, Exercise 16.8 provides a key regularity condition. More generally, one should be aware that the relative ease of implementing the Gibbs sampler in some problems does not obviate the need for careful mathematical analysis of the problem structure and the results. The last of the three examples here illustrates the derivation of the full conditional for a particular trivariate model used in spatial modeling.

Example 16.4—Gibbs sampling for a normal distribution. Suppose that there is interest in generating samples $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for some mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. As mentioned above for M-H, the Gibbs sampler may not be the most efficient method of generating samples from a multivariate normal distribution. Nevertheless, this example is useful as an illustration of the process of constructing the full conditionals.

A standard result from multivariate normality is that the distribution of any selection of components within $\boldsymbol{X}$ conditioned on the remaining components is also normal (e.g., Mardia et al., 1979, pp. 62–63). Specifically, the distribution of the ith component conditioned on the remaining components provides the sampling distribution:

$$\{X_{ki}|\boldsymbol{X}_{k\setminus i}\} \sim N\left(\mu_i + \boldsymbol{\Sigma}_{i,\setminus i}^T \boldsymbol{\Sigma}_{\setminus i,\setminus i}^{-1}(\boldsymbol{X}_{k\setminus i} - \boldsymbol{\mu}_{\setminus i}),\ \sigma_i^2 - \boldsymbol{\Sigma}_{i,\setminus i}^T \boldsymbol{\Sigma}_{\setminus i,\setminus i}^{-1} \boldsymbol{\Sigma}_{i,\setminus i}\right), \qquad (16.9)$$

where μ_i and σ_i^2 are, respectively, the ith component of $\boldsymbol{\mu}$ and ith diagonal component of $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}_{\setminus i}$ is the vector containing all components of $\boldsymbol{\mu}$ except μ_i, $\boldsymbol{\Sigma}_{i,\setminus i}$ is the column vector of the elements of $\boldsymbol{\Sigma}$ corresponding to the covariances between the ith component of $\boldsymbol{X}$ and all *other* components of $\boldsymbol{X}$, and $\boldsymbol{\Sigma}_{\setminus i,\setminus i}$ contains the elements of $\boldsymbol{\Sigma}$ with the row and column corresponding to the ith component of $\boldsymbol{X}$ removed. (Note that $\boldsymbol{\mu}_{\setminus i}$ and $\boldsymbol{\Sigma}_{i,\setminus i}$ are $(m-1)$-dimensional and $\boldsymbol{\Sigma}_{\setminus i,\setminus i}$ is $(m-1)\times(m-1)$-dimensional.) Hence, to generate sample values of $\boldsymbol{X}$ via the Gibbs sampler, the densities $p_i(x|\boldsymbol{X}_{k\setminus i})$, $i = 1, 2,\ldots, m$, in steps 1 and 3 of the procedure previously outlined are equal to the right-hand side of (16.9). ❑

Example 16.5—Gibbs sampling for truncated exponential distributions. Following Casella and George (1992), let X and Y have conditional exponential probability density functions on an interval $(0, B)$:

$$p_{X|Y}(x\,|\,y) = \frac{ye^{-yx}}{1-e^{-By}}, \quad 0 < x < B,$$

$$p_{Y|X}(y\,|\,x) = \frac{xe^{-xy}}{1-e^{-Bx}}, \quad 0 < y < B.$$

With the sampling densities $p_1(\cdot) = p_{X|Y}(\cdot)$ and $p_2(\cdot) = p_{Y|X}(\cdot)$, the Gibbs algorithm can be used to produce samples from the joint density $p_{X,Y}(x,y)$. It is easy to generate from these sampling densities using the inverse-transform method (Section D.2 of Appendix D). Suppose for the application here that the interest is in the marginal density $p_X(x)$ rather than the joint density.

As noted in Casella and George (1992), the density $p_X(x)$ does not exist for $B = \infty$ (see also Exercise 16.8). That is, at $B = \infty$, the marginal "density" that results is improper in the sense that $\int_0^\infty\left[\int_0^\infty p_{X,Y}(x,y)dy\right]dx = \infty$ even though both conditional densities above exist and are proper. From Robert and Casella (1999, Sect. 7.1.5), the existence of the joint density in this problem (from which the marginal for X can be determined) requires that $\int_0^B p_{Y|X}(y\,|\,x)/p_{X|Y}(x\,|\,y)\,dy < \infty$. As seen in Exercise 16.8, this condition is violated for $B = \infty$.

Nontrivial calculations show that for the truncated $B < \infty$ case, the marginal density exists and satisfies

$$p_X(x) = c\frac{1-e^{-Bx}}{x},$$

where c is the normalizing constant (Casella and George, 1992). Using the property $\int_0^B p_X(x)dx = 1$ and letting $B = 5$, it is found that $c \approx 0.2634$. This known marginal density can be used for comparisons with the output of the Gibbs sampler. (In practice, of course, the marginal or joint densities are usually unknown.)

Figure 16.2 shows a histogram of output for a Gibbs sampler based on $n = 40$. The histogram is constructed from the terminal output of the chain using 5000 independent replications, with each replication being initialized at $Y_0 = 2.5$ (from which X_1, Y_1, X_2, Y_2, etc. are generated). The histogram closely matches the marginal density, indicating that the chain output has a distribution close to the desired distribution.

We also tested the ability of the chain to provide an ergodic average close to the true value. Based on n above, let the burn-in period be $M = 10$. The chain is used to estimate the mean $E(X)$; that is, $f(X, Y) = X$ as in step 4 of the Gibbs sampling procedure above. For comparison purposes, the true mean is given by

$$E(X) = 0.2634\int_0^5 x\frac{1-e^{-5x}}{x}dx = 1.264.$$

Let us compare results based on the ergodic average without and with burn-in ($n = 40$, $M = 0$ and $n = 40$, $M = 10$, respectively). Based on 120 independent replications (using the M-file **Gibbs** available at the book's Web site), it is found that the mean of the 120 estimates for $E(X)$ is 1.296 without burn-in and 1.254 with burn-in. The corresponding sample standard deviations of the estimates are 0.259 and 0.294, respectively (so the sample standard deviations of the *means* of the 120 estimates are 0.0236 and 0.0268). In comparing these estimates relative to the true $E(X) = 1.264$, the resulting one-sample t-statistics are 1.356 for the no-burn-in implementation and 0.373 for the burn-in implementation. These values indicate that there is no evidence that either of the no-burn-in or burn-in implementations provides an estimate of $E(X)$ significantly different from the true value of 1.264 (i.e., the P-values are larger than 0.10). More detailed analysis would be required to determine if there is a significant difference between the no-burn-in and burn-in implementation. Note that the burn-in period (M) and overall run length (n) for this problem are shorter than needed in many other problems, where these numbers may easily run into the 100s or 1000s. ❑

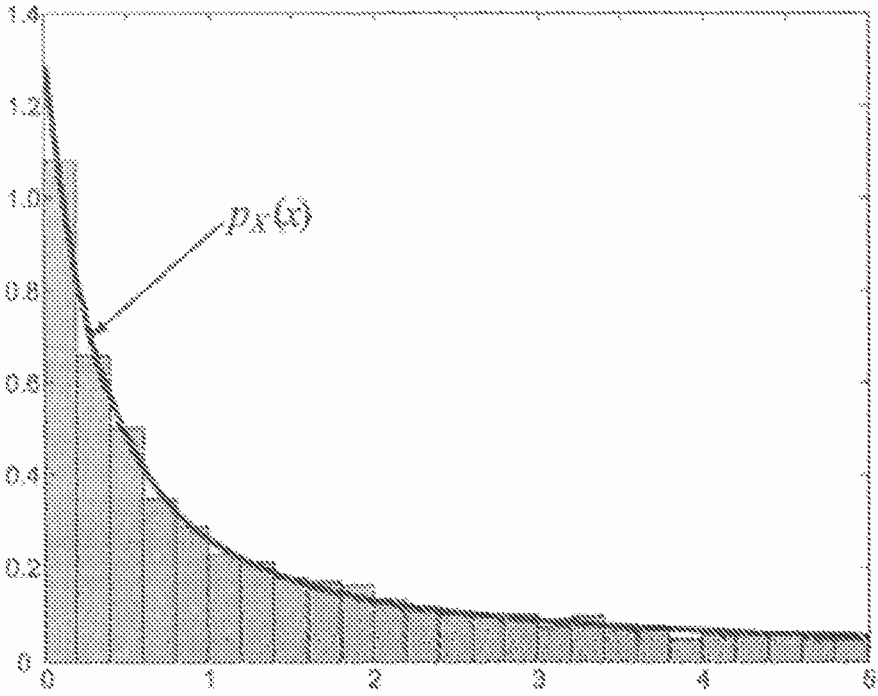

Figure 16.2. Histogram for terminal X from Gibbs sampling process with truncated exponential sampling. Histogram constructed from 5000 independent replications. The desired marginal density is shown by the solid line.

Example 16.6—Autoexponential model. The trivariate autoexponential model in Besag et al. (1995) has applications in some aspects of spatial modeling. The relevant random variables are represented by X, Y, Z, each defined on the interval $(0, \infty)$. The trivariate autoexponential density has the form

$$p_{X,Y,Z}(x, y, z) = c\exp[-(x + y + z + \theta_{xy}xy + \theta_{yz}yz + \theta_{xz}xz)],$$

where c is the normalizing constant and θ_{xy}, θ_{yz}, and θ_{xz} are known parameters. The density $p_{X,Y,Z}(x, y, z)$ represents the target density. There is no easy way to directly sample from $p_{X,Y,Z}(x, y, z)$. We are interested in constructing the full conditionals to be used in generating samples from this target density.

The Gibbs sampling procedure can be easily implemented since the three full conditional densities are simple scalar exponential densities (Appendix D shows how exponential random variables can be easily generated). For example, the full conditional for the random variable $\{Z|X{=}x, Y{=}y\}$ is an exponential density function with parameter $1+\theta_{yz}y+\theta_{xz}x$ (i.e., the mean of $\{Z|X{=}x, Y{=}y\}$ is $(1+\theta_{yz}y+\theta_{xz}x)^{-1}$; see Exercise 16.10). The two other full conditionals have analogous convenient exponential forms. Note that the derivation for these full conditionals does not depend on the normalizing constant c (due to cancellation via (16.7)). This is a critical advantage to sampling from the full conditionals. Note that *other* conditional and unconditional densities deriving from the autoexponential density above are not so convenient. For example, Robert and Casella (1999, p. 287) present the densities for $\{Y|X=x\}$ and X, both of which are nonexponential and cannot be used to easily generate samples. ❑

16.6 APPLICATIONS IN BAYESIAN ANALYSIS

The above discussion of MCMC has been for general problems where the desire is to obtain quantities related to a target distribution for a random vector $\boldsymbol{X}$. More specifically, MCMC—particularly Gibbs sampling—has had an especially profound impact on *Bayesian* methods of analysis. There seem to be at least two reasons for this: (i) The structure of Bayes' rule is well matched to the requirements of MCMC for drawing samples from appropriate conditional densities, and (ii) MCMC fills a long-standing need for a general-purpose method for constructing quantities related to posterior distributions that does not require cumbersome numerical integration. The seminal paper related to Bayesian applications of MCMC is Gelfand and Smith (1990).

Let us review the Bayesian framework. Suppose that $\boldsymbol{\Pi}$ represents a vector of terms important to the analysis of some system and that data $\boldsymbol{Z}$ can be collected on that system. For example, $\boldsymbol{\Pi}$ might be the parameters $\boldsymbol{\theta}$ of a model that are to be estimated; in an example below, $\boldsymbol{\Pi}$ represents both model parameters and the response to a varying input factor. In the Bayesian approach, $\boldsymbol{\Pi}$ is treated as a random vector instead of a fixed constant. Let us suppose that $\boldsymbol{\Pi}$ has a probability density, say $p_{\Pi}(\boldsymbol{\pi})$. (The general Bayesian approach can work with discrete or hybrid $\boldsymbol{\Pi}$ as well.) The density $p_{\Pi}(\boldsymbol{\pi})$ is referred to as the *prior density*, as it reflects a priori information available about $\boldsymbol{\Pi}$. There are many issues associated with the prior—philosophical and mathematical—but we will not delve into those here.

Bayes' rule takes the prior information on $\boldsymbol{\Pi}$, expressed via the prior density function, combines it with the conditional density function for the data $p_{Z|\Pi}(z|\boldsymbol{\pi})$ (sometimes called the *likelihood function*), and forms the *posterior density function* $p_{\Pi|Z}(\boldsymbol{\pi}|z)$ according to

$$p_{\Pi|Z}(\boldsymbol{\pi}\,|\,z)=\frac{p_{Z|\Pi}(z\,|\,\boldsymbol{\pi})p_{\Pi}(\boldsymbol{\pi})}{\int p_{Z|\Pi}(z\,|\,\boldsymbol{\pi})p_{\Pi}(\boldsymbol{\pi})d\boldsymbol{\pi}}=\frac{p_{\Pi,Z}(\boldsymbol{\pi},z)}{p_{Z}(z)}, \tag{16.10}$$

where the integral is over the domain for $\boldsymbol{\Pi}$. This simple-looking formula has profound implications and applications relative to modern learning and statistical analysis. The posterior density provides a fundamental means of characterizing the system parameters (or other quantities) $\boldsymbol{\Pi}$ by combining data ($\boldsymbol{Z}$) with prior information.

In almost all practical applications, numerical methods will be needed to calculate the integrals required for forming and using the posterior density. Rarely will the integration be feasible in closed form. One area requiring integration is the computation of the conditional expectation, $E[f(\boldsymbol{\Pi})|\boldsymbol{Z}] = \int f(\boldsymbol{\pi})p_{\Pi|Z}(\boldsymbol{\pi}|\boldsymbol{Z})d\boldsymbol{\pi}$, particularly the special case of the conditional mean $E[\boldsymbol{\Pi}|\boldsymbol{Z}]$. Further, in using the posterior density function in (16.10) to construct posterior *probabilities* associated with $\boldsymbol{\Pi}$, multifold integration of the left side of (16.10) is required. In particular, there might be interest in probabilities of the form $P(\boldsymbol{\Pi} \in S|\boldsymbol{Z}) = \int_S p_{\Pi|Z}(\boldsymbol{\pi}|\boldsymbol{Z})d\boldsymbol{\pi}$, where S is some subset of the domain for $\boldsymbol{\Pi}$. (This type of integration arises, for instance, in computing credible regions for $\boldsymbol{\Pi}$, the Bayesian analogue of confidence regions.) A final aspect involving multivariate integration is the computation of the marginal density for $\boldsymbol{Z}$ (i.e., the denominator of (16.10)). This marginal is needed in some applications (e.g., Chib, 1995), but (happily!) not in the Gibbs sampling for producing samples from the posterior, as we discuss below.

As mentioned at the beginning of this section, MCMC—particularly Gibbs sampling—is especially well suited to Bayesian analysis. Recall that the Gibbs sampling procedure can be implemented if one can construct full conditional densities for each element of the vector of interest and samples can be generated from the full conditionals. Following the discussion of Section 16.3, $\{\boldsymbol{\Pi}_{k+1,i}|\boldsymbol{\Pi}_{k\backslash i},\boldsymbol{Z}\} \sim p_i(\boldsymbol{\pi}|\boldsymbol{\Pi}_{k\backslash i},\boldsymbol{Z})$, $i = 1, 2,\ldots, m$, where the notation is analogous to the generic notation of Section 16.3. In particular, $\boldsymbol{\Pi}_k \equiv [\boldsymbol{\Pi}_{k1}, \boldsymbol{\Pi}_{k2},\ldots,\boldsymbol{\Pi}_{km}]^T$ and the set of $m-1$ components without the ith component is $\boldsymbol{\Pi}_{k\backslash i} \equiv \{\boldsymbol{\Pi}_{k+1,1}, \boldsymbol{\Pi}_{k+1,2},\ldots, \boldsymbol{\Pi}_{k+1,i-1}, \boldsymbol{\Pi}_{k,i+1},\ldots, \boldsymbol{\Pi}_{km}\}$. In the special case where $\boldsymbol{\Pi} = \boldsymbol{\theta}$, then $m = \dim(\boldsymbol{\theta})$ $(= p$, following previous notation) and the ith component of $\boldsymbol{\Pi}$ corresponds to the ith element of $\boldsymbol{\theta}$. As mentioned earlier, the individual components $\boldsymbol{\Pi}_{ki}$ may be scalar or multivariate. Note the extra conditioning ($\boldsymbol{Z}$) due to the data, reflecting the posterior aspect of the Bayesian formulation. This conditioning—although critical to the Bayesian analysis—may be treated as a constant relative to the Gibbs sampling process. We emphasize this treatment as a constant by writing $\boldsymbol{Z} = \boldsymbol{z}$ in the conditioning arguments below.

Expression (16.7) provides the fundamental form for the full conditionals. Substituting the right-hand side of (16.10) into (16.7) yields the full conditionals for the random variables,

$$p_i(\boldsymbol{\pi}|\boldsymbol{\Pi}_{k\backslash i},\boldsymbol{Z}=\boldsymbol{z}) = \frac{p_{\Pi|Z}(\boldsymbol{\pi},\boldsymbol{\Pi}_{k\backslash i}|\boldsymbol{z})}{\int p_{\Pi|Z}(\boldsymbol{\pi},\boldsymbol{\Pi}_{k\backslash i}|\boldsymbol{z})d\boldsymbol{\pi}} = \frac{p_{\Pi,Z}(\boldsymbol{\pi},\boldsymbol{\Pi}_{k\backslash i},\boldsymbol{z})}{\int p_{\Pi,Z}(\boldsymbol{\pi},\boldsymbol{\Pi}_{k\backslash i},\boldsymbol{z})d\boldsymbol{\pi}}, \quad (16.11)$$

where the second equality follows by the cancellation of the marginal density $p_Z(\boldsymbol{z})$ in the numerator and denominator. Hence, to create samples from the posterior density function via the Gibbs sampler it is *not* necessary to compute the denominator integral in Bayes' rule (16.10). In another context, we saw this desirable property in Example 16.6.

Example 16.7 sets up a Gibbs sampling implementation for the popular *variance-components model* (sometimes called the *random-effects model*). This example is considered in Gelfand and Smith (1990). The variance-components model is popular in engineering, clinical trials, and survey design as a means of studying systems where multiple input factors may have multiple levels.

Example 16.7—Variance-components model. Suppose that data arrive according to $Z_{ij} = \beta_i + \varepsilon_{ij}$, $i = 1, 2,\ldots, N_I$, $j = 1, 2,\ldots, N_J$, where $\{\beta_i \,|\, \mu,\sigma_\beta^2\} \sim N(\mu,\sigma_\beta^2)$ and $\{\varepsilon_{ij} \,|\, \sigma_\varepsilon^2\} \sim N(0,\sigma_\varepsilon^2)$. Here, Z_{ij} represents the measured response of the *j*th replication of the *i*th level of the factor. The term β_i represents the underlying random response to the *i*th level of the factor, and the noise ε_{ij} represents the measurement error for the *j*th replication of the *i*th level of the factor. Given that $\{\beta_i \,|\, \mu,\sigma_\beta^2\}$ and $\{\varepsilon_{ij} \,|\, \sigma_\varepsilon^2\}$ are independent, it follows that $\{Z_{ij} \,|\, \beta_i,\sigma_\varepsilon^2\} \sim N(\beta_i,\sigma_\varepsilon^2)$. Let $\boldsymbol{Z} = [Z_{11},\ldots, Z_{1,N_J}; Z_{21},\ldots; Z_{N_I,1},\ldots, Z_{N_I N_J}]^T$ and $\boldsymbol{\beta} = [\beta_1, \beta_1,\ldots, \beta_{N_I}]^T$.

In the Bayesian context, we take the parameters μ, σ_β^2, and σ_ε^2 as random (and independent). The prior distributions for μ, σ_β^2, and σ_ε^2 are $N(\mu_0,\sigma_0^2)$, $\text{IG}(a_\beta, b_\beta)$, and $\text{IG}(a_\varepsilon, b_\varepsilon)$, respectively, where μ_0,σ_0^2, a_β, b_β, a_ε, and b_ε are known parameters of the prior densities and $\text{IG}(\cdot)$ denotes an inverse gamma distribution. (For $\text{IG}(a, b)$, the density with argument σ^2 is proportional to $(\sigma^2)^{-a-1}\exp(-b/\sigma^2)$; the constant of proportionality depends on a, b and the gamma function, as shown in Gelfand and Smith, 1990, and Chen et al., 2000, p. 244.)

Putting the above pieces together, the joint density for $[\Pi; \boldsymbol{Z}] = [\boldsymbol{\beta}, \mu, \sigma_\beta^2, \sigma_\varepsilon^2; \boldsymbol{Z}]$ is

$$\{\boldsymbol{Z} \,|\, \boldsymbol{\beta},\sigma_\varepsilon^2\} * \{\boldsymbol{\beta} \,|\, \mu,\sigma_\beta^2\} * \{\mu\} * \{\sigma_\beta^2\} * \{\sigma_\varepsilon^2\}$$

$$= N(\boldsymbol{\beta},\sigma_\varepsilon^2 \boldsymbol{I}_{N_I N_J}) * N(\mu \mathbf{1}_{N_I},\sigma_\beta^2 \boldsymbol{I}_{N_I}) * N(\mu_0,\sigma_0^2) * \text{IG}(a_\beta, b_\beta) * \text{IG}(a_\varepsilon, b_\varepsilon), \tag{16.12}$$

where the $*$ operator denotes the multiplication of the density functions associated with (as appropriate) the indicated random variables or the indicated distributions, and $\mathbf{1}_{N_I}$ denotes an N_I-dimensional vector of 1's. The density in (16.12) is the numerator in Bayes' rule (16.10). For our purposes, the four terms in $\Pi = [\boldsymbol{\beta}, \mu, \sigma_\beta^2, \sigma_\varepsilon^2]$ are of interest. Note that the first component of Π (i.e., $\boldsymbol{\beta}$) is multivariate, illustrating the point above that for some applications it is beneficial to have multivariate components in forming full conditionals.

We can use the following four full conditionals to construct samples from the posterior $p_{\Pi|Z}(\pi|z)$. As given in Gelfand and Smith (1990), the full conditionals are built from the joint density in (16.12) according to (16.11):

$$\{\boldsymbol{\beta} \mid \mu, \sigma_\beta^2, \sigma_\varepsilon^2, \boldsymbol{Z}\}$$
$$\sim N\left(\frac{N_J\sigma_\beta^2}{N_J\sigma_\beta^2 + \sigma_\varepsilon^2}\bar{\boldsymbol{Z}} + \frac{\mu\sigma_\varepsilon^2}{N_J\sigma_\beta^2 + \sigma_\varepsilon^2}\mathbf{1}_{N_I},\ \frac{\sigma_\beta^2\sigma_\varepsilon^2}{N_J\sigma_\beta^2 + \sigma_\varepsilon^2}\boldsymbol{I}_{N_I} \right),$$

$$\{\mu \mid \boldsymbol{\beta}, \sigma_\beta^2, \sigma_\varepsilon^2, \boldsymbol{Z}\} = \{\mu \mid \boldsymbol{\beta}, \sigma_\beta^2\}$$
$$\sim N\left(\frac{\sigma_\beta^2\mu_0 + \sigma_\beta^2\sum_i \beta_i}{N_I\sigma_\beta^2 + \sigma_\beta^2},\ \frac{\sigma_\beta^2\sigma_\varepsilon^2}{N_I\sigma_\beta^2 + \sigma_\beta^2} \right),$$

$$\{\sigma_\beta^2 \mid \boldsymbol{\beta}, \mu, \sigma_\varepsilon^2, \boldsymbol{Z}\} = \{\sigma_\beta^2 \mid \boldsymbol{\beta}, \mu\}$$
$$\sim \text{IG}\left(a_\beta + \tfrac{1}{2}N_I,\ b_\beta + \tfrac{1}{2}\textstyle\sum_i (\beta_i - \mu)^2\right),$$

$$\{\sigma_\varepsilon^2 \mid \boldsymbol{\beta}, \mu, \sigma_\beta^2, \boldsymbol{Z}\} = \{\sigma_\varepsilon^2 \mid \boldsymbol{\beta}, \boldsymbol{Z}\}$$
$$\sim \text{IG}\left(a_\varepsilon + \tfrac{1}{2}N_I N_J,\ b_\varepsilon + \tfrac{1}{2}\textstyle\sum_i\sum_j (Z_{ij} - \mu)^2\right),$$

where $\bar{\boldsymbol{Z}} = N_J^{-1}[\sum_j Z_{1j}, \ldots, \sum_j Z_{N_I j}]^T$. It is relatively easy to sample from the above four distributions in the cyclic manner of the Gibbs algorithm. In particular, the sampling densities, $p_i(\pi \mid \Pi_{k\backslash i}, \boldsymbol{Z})$, $i = 1, 2, 3, 4$, correspond to the four full conditionals above. Hence, provided that the sampler has run long enough and/or that burn-in effects are removed, the easy sampling of the full conditionals provides a means for estimating quantities from the otherwise formidable $p_{\Pi|Z}(\pi|z)$. ❑

As we have seen with so many other aspects of stochastic search, there are connections between root-finding stochastic approximation (SA) (Chapter 4) and MCMC as well. Gu and Kong (1998) and Gu and Zhu (2001) show how maximum likelihood estimates can be computed via SA, where MCMC methods are used to approximate the gradient and Hessian matrix of the log-likelihood function. Hence, the MCMC-based calculations provide the noisy input that fits into the classical SA framework (analogous to $\boldsymbol{Y}(\boldsymbol{\theta})$ in Chapter 4 and elsewhere). Chen et al. (2000, pp. 323–325) summarizes an application of SA in finding the mode of unimodal posterior distributions for a subset of parameters of interest. In this application, one is interested in solving the root-finding problem:

$$\frac{\partial \log p_{\Pi|Z}}{\partial \boldsymbol{\theta}_{(1)}} = \mathbf{0},$$

where $\Pi = [\boldsymbol{\theta}_{(1)}, \boldsymbol{\theta}_{(2)}]$ represents a collection of model parameters with $\boldsymbol{\theta}_{(1)}$ representing the parameters of interest and $\boldsymbol{\theta}_{(2)}$ representing other parameters. A standard root-finding SA recursion may be applied to estimate $\boldsymbol{\theta}_{(1)}$; the values for $\boldsymbol{\theta}_{(2)}$ that are required to get the noisy measurement of the gradient above (for use in the SA algorithm) are generated via MCMC. This is a convenient way of getting the required noisy gradient estimate $\boldsymbol{Y}(\boldsymbol{\theta})$.

One of the areas of application for Bayesian-based Gibbs sampling is state and parameter estimation in dynamic (state-space) models (see Subsection 3.3.4 for a summary of such models). The Gibbs sampler allows for the treatment of nonstandard conditions, such as nonlinearity for the system dynamics and/or nonnormality for the noise terms. In such conditions, the Kalman filter may yield poor results or may be inapplicable. A summary of the overall approach and some pointers to the broader literature is given in Spall (2003).

16.7 CONCLUDING REMARKS

The discussion above summarizes important aspects of Markov chain Monte Carlo, including the motivation, theory, implementation, and connection to Bayesian analysis. The focus is on the Metropolis–Hastings and Gibbs sampling versions of MCMC. Many large-scale practical implementations of MCMC borrow aspects from both M-H and Gibbs sampling.

Although Gibbs may be considered a component-wise M-H, the techniques have developed largely independent of each other. Recognizing this, we discuss M-H and Gibbs as separate approaches. As with other stochastic search methods, no one approach is to be universally preferred. One strong aspect of both M-H and Gibbs is the theory supporting the methods and guaranteeing convergence under modest conditions.

The M-H method is more general than Gibbs sampling. The Gibbs sampler has the relatively strong requirement that the full conditional distributions be available. M-H has no such requirement and in fact provides almost complete flexibility in the choice of the distribution from which to simulate. Assuming that the full conditional distributions are available for simulation, the Gibbs sampler has more intuitive appeal than the M-H algorithm. Building up the joint distribution from the set of full conditionals does not seem to require the "leap of logic" that is needed with the M-H algorithm and its arbitrary choice of sampling (proposal) distribution (although M-H *is* on a fully sound theoretical footing). Bayesian problems provide a framework especially well suited to Gibbs sampling as a result of the frequent availability of the full conditionals.

Because the structure of Gibbs sampling is more restrictive than M-H, Gibbs has the advantage of not needing the tuning that is required in M-H. A serious application of M-H will usually require some experimentation with the proposal distribution, not only in the specific parameters of the distribution, but perhaps in the general *form* of the distribution (e.g., gamma or uniform?). The restrictions in Gibbs sampling are analogous to certain search methods (e.g., Newton–Raphson in Section 1.4 or random search algorithm A in Section 2.2), where the tuning is eliminated because the structure of the algorithm is sufficiently proscribed.

It is not possible to say which of M-H or Gibbs is computationally more efficient in general. Much depends on the specifics of the implementation. Examples are available in the literature illustrating both efficient and inefficient results for either or both approaches.

We saw, for instance, a highly efficient Gibbs implementation in the truncated exponential problem of Example 16.5 (i.e., very few iterations required to obtain samples having the required distribution). This is consistent with the general principle that Gibbs—like other methods in other settings—may benefit by taking into account the structure of the problem. Gibbs also has the advantage of low-dimensional—usually *scalar*—random number generation for arbitrarily high-dimensional problems. On the other hand, the component-wise generation in Gibbs sampling may produce a phenomenon analogous to multivariate optimization with only one component at a time. In optimization, this may lead to convergence to a saddlepoint or local optimum; in Gibbs sampling, this may lead to slow exploration of the space of possible outcomes for $f(X)$, leading to a poor estimate for $E[f(X)]$. Robert and Casella (1999, pp. 318–319) illustrate slow convergence via this phenomenon in a problem involving a mixture distribution (a mixture of normals). In M-H, slow convergence is generally associated with a poor choice of the proposal distribution relative to the target distribution. This may cause M-H to miss some of the finer structure of the target distribution. There are also highly efficient M-H implementations, especially when the algorithm benefits by some tuning (e.g., Chib and Greenberg, 1995; Robert and Casella, 1999, Chap. 6).

The ultimate value of M-H or Gibbs sampling, of course, is their usefulness in solving practical problems. Both approaches have proved to be important tools in the modern analyst's toolbox.

EXERCISES

16.1 Discuss why the ergodic average of a given number of samples in typical applications of the M-H (and other) algorithms will have greater variability than a corresponding average of the same number of independent samples of $f(X)$. In the demonstration of this point in Example 16.1, verify analytically that the standard deviation for the *independent samples* case is 0.0159.

16.2 (a) Based on 50 independent replications of the M-H algorithm in Example 16.1 with the uniform proposal distribution used in Figure 16.1, test whether the mean of the terminal estimate is statistically indistinguishable from the true value of zero.

(b) Repeat the test with a normal proposal distribution but other aspects of Example 16.1 unchanged. In particular, assume that $\{\boldsymbol{W}|\boldsymbol{X}=\boldsymbol{x}\} \sim N(\boldsymbol{x}, \boldsymbol{I}_2/12)$ (note that this proposal distribution has the same mean and variance as the original uniform distribution).

(c) Finally, do the same test above, but with a $U_2(\boldsymbol{x} - 2\mathbf{1}_2, \boldsymbol{x} + 2\mathbf{1}_2)$ proposal distribution (see Table 16.1). Comment on the observed differences in the performance for the three proposal distributions.

16.3 Consider the setting of Example 16.1 except that the off-diagonal (covariance) term in the covariance matrix for $\boldsymbol{X}$ changes from 0.9 to 0.5. Based on a $U_2(\boldsymbol{x} - 3\mathbf{1}_2, \boldsymbol{x} + 3\mathbf{1}_2)$ proposal distribution, produce a plot of four independent replications of estimates for $E[f(\boldsymbol{X})]$ over the range of 0 to 10,000 post-burn-in iterations (analogous to Figure 16.1) and determine the approximate acceptance rate for the candidate points $\boldsymbol{W}$. Use the burn-in period and $\boldsymbol{X}_0$ of Example 16.1.

16.4 Consider a special case of M-H where $q(\boldsymbol{w}|\boldsymbol{x}) = q(\boldsymbol{w})$. Suppose that there exists a $C \geq 1$ such that $p(\boldsymbol{x}) \leq Cq(\boldsymbol{x})$ for all $\boldsymbol{x}$ in the support of $p(\boldsymbol{x})$. Let $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ be continuous functions. Prove that the expected acceptance probability at any iteration (i.e., $E[\rho(\boldsymbol{X}, \boldsymbol{W})]$) is at least $1/C$. (Hint: The "Answers to Selected Exercises" section near the end of the book provides an outline of the proof.) How does this bound relate to the acceptance probability for the accept–reject method (Appendix D)?

16.5 In Example 16.3, derive the values for the transition matrix $\boldsymbol{P}$ and verify that the balance equation $\bar{\boldsymbol{p}}^T = \bar{\boldsymbol{p}}^T\boldsymbol{P}$ holds.

16.6 Suppose that X is bivariate normally distributed where the marginal distribution for the two components is $N(0, 1)$ and the correlation is ρ. Present the two sampling distributions, $p_1(x|X_{k2})$ and $p_2(x|X_{k1})$, for use in step 1 (and 3) of the Gibbs sampling algorithm.

16.7 Consider the setting of Example 16.5 with the exception that the Gibbs sampler is now used to estimate the *second moment* of X (relative to $p_X(x)$) rather than the mean (first moment) of X. Assume the same burn-in period, number of replications, and initial condition for Y. Compute the true second moment and test for a bias in the estimate of the second moment when using an ergodic average with no burn-in ($M = 0$) and when using the specified burn-in ($M = 10$).

16.8 As discussed in Example 16.5, the existence of the full conditional densities does not guarantee the existence of the joint density $p_{X,Y}(x, y)$. In this two-variable problem, the existence of the target density requires that

$$\int \frac{p_{Y|X}(y\,|\,x)}{p_{X|Y}(x\,|\,y)}\,dy < \infty,$$

where the integral is over the domain for Y. Show that this condition is violated for Example 16.5 when $B = \infty$.

16.9 Consider the setting of Example 16.5 except that $B = 4$. Determine the normalizing constant c and produce a plot of the true density for X and corresponding histogram (analogous to Figure 16.2) based on 2000 independent replications and the terminal output at $n = 500$.

16.10 For the setting of Example 16.6, establish that the full conditional density function for $\{Z\,|\,X = x,\ Y = y\}$ has the exponential form given in the example.