Markov Chain Monte Carlo Methods

Introduction

It is, in general, very difficult to simulate the value of a random vector **X** whose component random variables are dependent. In this chapter we present a powerful approach for generating a vector whose distribution is approximately that of **X**. This approach, called the Markov chain Monte Carlo method, has the added significance of only requiring that the mass (or density) function of **X** be specified up to a multiplicative constant, and this, we will see, is of great importance in applications.

In Section 12.1 we introduce and give the needed results about Markov chains. In Section 12.2 we present the Hastings–Metropolis algorithm for constructing a Markov chain having a specified probability mass function as its limiting distribution. A special case of this algorithm, referred to as the Gibbs sampler, is studied in Section 12.3. The Gibbs sampler is probably the most widely used Markov chain Monte Carlo method. An application of the preceding methods to deterministic optimization problems, known as simulated annealing, is presented in Section 12.5. In Section 12.6 we present the sampling importance resampling (SIR) technique. While not strictly a Markov chain Monte Carlo algorithm, it also results in approximately simulating a random vector whose mass function is specified up to a multiplicative constant.

12.1 Markov Chains

Consider a collection of random variables $X_0, X_1, \ldots$. Interpret X_n as the "state of the system at time n," and suppose that the set of possible values of the X_n—that is, the possible states of the system—is the set $1, \ldots, N$. If there exists a set of numbers $P_{ij}, i, j = 1, \ldots, N$, such that whenever the process is in state i then,

Simulation. DOI: http://dx.doi.org/10.1016/B978-0-12-415825-2.00012-7

independent of the past states, the probability that the next state is j is P_{ij}, then we say that the collection $\{X_n, n \geq 0\}$ constitutes a *Markov chain* having transition probabilities $P_{ij}, i, j = 1, \ldots, N$. Since the process must be in some state after it leaves states i, these transition probabilities satisfy

$$\sum_{j=1}^{N} P_{ij} = 1, \quad i = 1, \ldots, N$$

A Markov chain is said to be irreducible if for each pair of states i and j there is a positive probability, starting in state i, that the process will ever enter state j. For an irreducible Markov chain, let π_j denote the long-run proportion of time that the process is in state j. (It can be shown that π_j exists and is constant, with probability 1, independent of the initial state.) The quantities π_j, $j = 1, \ldots, N$, can be shown to be the unique solution of the following set of linear equations:

$$\begin{aligned} \pi_j &= \sum_{i=1}^{N} \pi_i P_{ij}, \quad j = 1, \ldots, N \\ \sum_{j=1}^{N} \pi_j &= 1 \end{aligned} \tag{12.1}$$

Remark The set of Equations (12.2) have a heuristic interpretation. Since π_i is the proportion of time that the Markov chain is in state i and since each transition out of state i is into state j with probability P_{ij}, it follows that $\pi_i P_{ij}$ is the proportion of time in which the Markov chain has just entered state j from state i. Hence, the top part of Equation (12.2) states the intuitively clear fact that the proportion of time in which the Markov chain has just entered state j is equal to the sum, over all states i, of the proportion of time in which it has just entered state j from state i. The bottom part of Equation (12.2) says, of course, that summing the proportion of time in which the chain is in state j, over all j, must equal 1. □

The $\{\pi_j\}$ are often called the *stationary probabilities* of the Markov chain. For if the initial state of the Markov chain is distributed according to the $\{\pi_j\}$ then $P\{X_n = j\} = \pi_j$, for all n and j (see Exercise 1).

An important property of Markov chains is that for any function h on the state space, with probability 1,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} h(X_i) = \sum_{j=1}^{N} \pi_j h(j) \tag{12.2}$$

The preceding follows since if $p_j(n)$ is the proportion of time that the chain is in state j between times $1, \ldots, n$ then

$$\frac{1}{n} \sum_{i=1}^{n} h(X_i) = \sum_{j=1}^{N} h(j) p_j(n) \to \sum_{j=1}^{N} h(j) \pi_j$$

The quantity π_j can often be interpreted as the limiting probability that the chain is in state j. To make precise the conditions under which it has this interpretation, we first need the definition of an aperiodic Markov chain.

Definition *An irreducible Markov chain is said to be aperiodic if for some* $n \geq 0$ *and some state* j*,*

$$P\{X_n = j | X_0 = j\} > 0 \quad \text{and} \quad P\{X_{n+1} = j | X_0 = j\} > 0$$

It can be shown that if the Markov chain is irreducible and aperiodic then

$$\pi_j = \lim_{n \to \infty} P\{X_n = j\}, \quad j = 1, \ldots, N$$

There is sometimes an easier way than solving the set of Equations (12.1) of finding the stationary probabilities. Suppose one can find positive numbers x_j, $j = 1, \ldots, N$ such that

$$x_i P_{ij} = x_j P_{ji}, \quad \text{for } i \neq j, \quad \sum_{j=1}^{N} x_j = 1$$

Then summing the preceding equations over all states i yields

$$\sum_{i=1}^{N} x_i P_{ij} = x_j \sum_{i=1}^{N} P_{ji} = x_j$$

which, since $\{\pi_j, j = 1, \ldots, N\}$ is the unique solution of (12.1), implies that

$$\pi_j = x_j$$

When $\pi_i P_{ij} = \pi_j P_{ji}$, for all $i \neq j$, the Markov chain is said to be *time reversible*, because it can be shown, under this condition, that if the initial state is chosen according to the probabilities $\{\pi_j\}$, then starting at any time the sequence of states going backwards in time will also be a Markov chain with transition probabilities P_{ij}.

Suppose now that we want to generate the value of a random variable X having probability mass function $P\{X = j\} = p_j$, $j = 1, \ldots, N$. If we could generate an irreducible aperiodic Markov chain with limiting probabilities p_j, $j = 1, \ldots, N$, then we would be able to approximately generate such a random variable by running the chain for n steps to obtain the value of X_n, where n is large. In addition, if our objective was to generate many random variables distributed according to p_j, $j = 1, \ldots, N$, so as to be able to estimate $E[h(X)] = \sum_{j=1}^{N} h(j) p_j$, then we could also estimate this quantity by using the estimator $\frac{1}{n} \sum_{i=1}^{n} h(X_i)$. However, since the early states of the Markov chain can be strongly influenced by the initial state chosen, it is common in practice to disregard the first k states, for some suitably chosen value of k. That is, the estimator $\frac{1}{n-k}\sum_{i=k+1}^{n} h(X_i)$, is utilized. It is difficult to know exactly how large a value of k should be used [although the advanced reader should see Aarts and Korst (1989) for some useful results along this line] and usually one just uses one's intuition (which usually works fine because the convergence is guaranteed no matter what value is used).

An important question is how to use the simulated Markov chain to estimate the mean square error of the estimator. That is, if we let $\hat{\theta} = \frac{1}{n-k}\sum_{i=k+1}^{n} h(X_i)$, how do we estimate

$$\text{MSE} = E\left[\left(\hat{\theta} - \sum_{j=1}^{N} h(j)p_j\right)^2\right]$$

One way is the *batch means* method, which works as follows. Break up the $n-k$ generated states into s batches of size r, where $s = (n-k)/r$ is integral, and let Y_j, $j = 1, \ldots, s$ be the average of the jth batch. That is,

$$Y_j = \frac{1}{r}\sum_{i=k+(j-1)r+1}^{k+jr} h(X_i), \quad j = 1, \ldots, s$$

Now, treat the Y_j, $j = 1, \ldots, s$ as if they were independent and identically distributed with variance σ^2 and use their sample variance $\hat{\sigma}^2 = \sum_{j=1}^{s}(Y_j - \overline{Y})^2/(s-1)$ as the estimator of σ^2. The estimate of MSE is $\hat{\sigma}^2/s$. The appropriate value of r depends on the Markov chain being simulated. The closer $X_i, i \geq 1$, is to being independent and identically distributed, then the smaller should be the value of r.

In the next two sections we will show, for a given set of positive numbers b_j, $j = 1, \ldots, N$, how to construct a Markov chain whose limiting probabilities are $\pi_j = b_j / \sum_{i=1}^{N} b_i$, $j = 1, \ldots, N$.

12.2 The Hastings–Metropolis Algorithm

Let $b(j)$, $j = 1, \ldots, m$ be positive numbers, and $B = \sum_{j=1}^{m} b(j)$. Suppose that m is large and B is difficult to calculate, and that we want to simulate a random variable (or a sequence of random variables) with probability mass function

$$\pi(j) = b(j)/B, \quad j = 1, \ldots, m$$

One way of simulating a sequence of random variables whose distributions converge $\pi(j)$, $j = 1, \ldots, m$, is to find a Markov chain that is easy to simulate and whose limiting probabilities are the $\pi(j)$. The *Hastings–Metropolis algorithm* provides an approach for accomplishing this task. It constructs a time-reversible Markov chain with the desired limiting probabilities, in the following manner.

Let $\mathbf{Q}$ be an irreducible Markov transition probability matrix on the integers $1, \ldots, m$, with $q(i, j)$, representing the row i, column j element of $\mathbf{Q}$. Now define a Markov chain $\{X_n, n \geq 0\}$ as follows. When $X_n = i$, a random variable X such that $P\{X = j\} = q(i, j)$, $j = 1, \ldots, m$, is generated. If $X = j$, then X_{n+1} is set equal to j with probability $\alpha(i, j)$ and is set equal to i with probability $1 - \alpha(i, j)$. Under these conditions, it is easy to see that the sequence of states will constitute a Markov chain with transition probabilities $P_{i,j}$ given by

$$\begin{aligned} P_{i,j} &= q(i, j)\alpha(i, j), \quad \text{if } j \neq i \\ P_{i,i} &= q(i, i) + \sum_{k \neq i} q(i, k)(1 - \alpha(i, k)) \end{aligned}$$

Now this Markov chain will be time reversible and have stationary probabilities $\pi(j)$ if

$$\pi(i)P_{i,j} = \pi(j)P_{j,i} \quad \text{for } j \neq i$$

which is equivalent to

$$\pi(i)q(i, j)\alpha(i, j) = \pi(j)q(j, i)\alpha(j, i)$$

It is now easy to check that this will be satisfied if we take

$$\alpha(i, j) = \min\left(\frac{\pi(j)q(j, i)}{\pi(i)q(i, j)}, 1\right) = \min\left(\frac{b(j)q(j, i)}{b(i)q(i, j)}, 1\right) \tag{12.3}$$

[To check, note that if $\alpha(i, j) = \pi(j)q(j, i)/\pi(i)q(i, j)$ then $\alpha(j, i) = 1$, and vice versa.]

The reader should note that the value of B is not needed to define the Markov chain, as the values $b(j)$ suffice. Also, it is almost always the case that $\pi(j)$, $j = 1, \ldots, m$, will not only be stationary probabilities but will also be limiting probabilities. (Indeed, a sufficient condition is that $P_{i,i} > 0$ for some i.)

The following sums up the Hastings–Metropolis algorithm for generating a time-reversible Markov chain whose limiting probabilities are $\pi(j) = b(j)/B$, $j = 1, \ldots, m$.

1. Choose an irreducible Markov transition probability matrix $\mathbf{Q}$ with transition probabilities $q(i, j), i, j = 1, \ldots, m$. Also, choose some integer value k between 1 and m.
2. Let $n = 0$ and $X_0 = k$.
3. Generate a random variable X such that $P\{X = j\} = q(X_n, j)$ and generate a random number U.
4. If $U < [b(X)q(X, X_n)]/[b(X_n)q(X_n, X)]$, then $NS = X$; else $NS = X_n$.
5. $n = n + 1$, $X_n = NS$.
6. Go to 3.

Example 12a Suppose that we want to generate a random element from a large complicated "combinatorial" set ℓ. For instance, ℓ might be the set of all permutations $(x_1, \ldots, x_n)$ of the numbers $(1, \ldots, n)$ for which $\sum_{j=1}^{n} jx_j > a$ for a given constant a; or ℓ might be the set of all subgraphs of a given graph having the property that for any pair of vertices i and j there is a unique path in the subgraph from i to j (such subgraphs are called trees).

To accomplish our goal we will utilize the Hastings–Metropolis algorithm. We shall start by assuming that one can define a concept of "neighboring" elements of ℓ, and we will then construct a graph whose set of vertices is ℓ by putting an arc between each pair of neighboring elements in ℓ. For example, if ℓ is the set of permutations $(x_1, \ldots, x_n)$ for which $\sum_{j=1}^{n} jx_j > a$, then we can define two such permutations to be neighbors if one results from an interchange of two of the positions of the other. That is (1, 2, 3, 4) and (1, 2, 4, 3) are neighbors, whereas (1, 2, 3, 4) and (1, 3, 4, 2) are not. If ℓ is a set of trees, then we can say that two trees are neighbors if all but one of the arcs of one of the trees are also arcs of the other tree.

Assuming this concept of neighboring elements, we define the q transition probability function as follows. With $N(s)$ defined as the set of neighbors of s, and $|N(s)|$ equal to the number of elements in the set $N(s)$, let

$$q(s, t) = \frac{1}{|N(s)|}, \quad \text{if } t \in N(s)$$

That is, the target next state from s is equally likely to be any of its neighbors. Since the desired limiting probabilities of the Markov chain are $\pi(s) = C$, it follows that $\pi(s) = \pi(t)$, and so

$$\alpha(s, t) = \min(|N(s)|/|N(t)|, 1)$$

That is, if the present state of the Markov chain is s, then one of its neighbors is randomly chosen—say it is t. If t is a state with fewer neighbors than s (in graph theory language, if the degree of vertex t is less than that of vertex s), then the next state is t. If not, a random number U is generated, and the next state is t if $U < |N(s)|/|N(t)|$, and is s otherwise. The limiting probabilities of this Markov chain are $\pi(s) = 1/|\ell|$. □

12.3 The Gibbs Sampler

The most widely used version of the Hastings–Metropolis algorithm is the *Gibbs sampler*. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a random vector with probability mass function (or probability density function in the continuous case) $p(\mathbf{x})$ that need only be specified up to a multiplicative constant, and suppose that we want to generate a random vector whose distribution is that of $\mathbf{X}$. That is, we want to generate a random vector having mass function

$$p(\mathbf{x}) = Cg(\mathbf{x})$$

where $g(\mathbf{x})$ is known, but C is not. Utilization of the Gibbs sampler assumes that for any i and values x_j, $j \neq i$, we can generate a random variable X having the probability mass function

$$P\{X = x\} = P\{X_i = x | X_j = x_j, j \neq i\} \tag{12.4}$$

It operates by using the Hastings–Metropolis algorithm on a Markov chain with states $\mathbf{x} = (x_1, \ldots, x_n)$, and with transition probabilities defined as follows. Whenever the present state is $\mathbf{x}$, a coordinate that is equally likely to be any of $1, \ldots, n$ is chosen. If coordinate i is chosen, then a random variable X whose probability mass function is as given by Equation (12.4) is generated, and if $X = x$ then the state $\mathbf{y} = (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$ is considered as the candidate next state. In other words, with $\mathbf{x}$ and $\mathbf{y}$ as given, the Gibbs sampler uses the Hastings–Metropolis algorithm with

$$q(\mathbf{x}, \mathbf{y}) = \frac{1}{n} P\{X_i = x | X_j = x_j, j \neq i\} = \frac{p(\mathbf{y})}{nP\{X_j = x_j, j \neq i\}}$$

Because we want the limiting mass function to be p, we see from Equation (12.3) that the vector $\mathbf{y}$ is then accepted as the new state with probability

$$\begin{aligned}\alpha(\mathbf{x}, \mathbf{y}) &= \min\left(\frac{p(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})q(\mathbf{x}, \mathbf{y})}, 1\right)\\ &= \min\left(\frac{p(\mathbf{y})p(\mathbf{x})}{p(\mathbf{x})p(\mathbf{y})}, 1\right)\\ &= 1\end{aligned}$$

Hence, when utilizing the Gibbs sampler, the candidate state is always accepted as the next state of the chain.

Example 12b Suppose we want to generate n random points in the circle of radius 1 centered at the origin, conditional on the event that no two points are within a distance d of each other, where

$$\beta = P\{\text{no two points are within } d \text{ of each other}\}$$

is assumed to be a small positive number. (If β were not small, then we could just continue to generate sets of n random points in the circle, stopping the first time that no two points in the set are within d of each other.) This can be accomplished by the Gibbs sampler by starting with n points in the circle, $x_1, \ldots, x_n$, such that no two are within a distance d of each other. Then generate a random number U and let $I = \text{Int}(nU) + 1$. Also generate a random point in the circle. If this point is not within d of any of the other $n - 1$ points excluding $\boldsymbol{x}_I$, then replace $\boldsymbol{x}_I$ by this generated point; otherwise, generate a new point and repeat the operation. After a large number of iterations the set of n points will approximately have the desired distribution. □

Example 12c Queueing Networks Suppose that r individuals move among $m+1$ queueing stations, and let, for $i = 1, \ldots, m$, $X_i(t)$ denote the number of individuals at station i at time t. If

$$p(n_1, \ldots, n_m) = \lim_{t\to\infty} P\{X_i(t) = n_i, i = 1, \ldots, m\}$$

then, assuming exponentially distributed service times, it can often be established that

$$p(n_1, \ldots, n_m) = C \prod_{i=1}^{m} P_i(n_i), \quad \text{if} \sum_{i=1}^{m} n_i \leq r$$

where $P_i(n), n \geq 0$ is a probability mass function for each $i = 1, \ldots, m$. Such a joint probability mass function is said to have a *product form*.

Although it is often relatively straightforward both to establish that $p(n_1, \ldots, n_m)$ has the preceding product form and to find the mass functions P_i, it can be difficult to explicitly compute the constant C. For even though

$$C \sum_{\boldsymbol{n}\,:\,s(\boldsymbol{n})\leq r} \prod_{i=1}^{m} P_i(n_i) = 1$$

where $\boldsymbol{n} = (n_1, \ldots, n_m)$ and $s(\boldsymbol{n}) = \sum_{i=1}^{m} n_i$, it can be difficult to utilize this result. This is because the summation is over all nonnegative integer vectors $\boldsymbol{n}$ for which $\sum_{i=1}^{m} n_i \leq r$ and there are $\binom{r+m}{m}$ such vectors, which is a rather large number even when m and r are of moderate size.

Another approach to learning about $p(n_1, \ldots, n_m)$, which finesses the computational difficulties of computing C, is to use the Gibbs sampler to generate a sequence of values having a distribution approximately that of p.

To begin, note that if $N = (N_1, \ldots, N_m)$ has the joint mass function p, then, for $n = 0, \ldots, r - \sum_{k\neq i} n_k$,

$$\begin{aligned} P\{N_i &= n|N_1 = n_1, \ldots, N_{i-1} = n_{i-1}, N_{i+1} = n_{i+1}, \ldots, N_m = n_m\} \\ &= \frac{p(n_1, \ldots, n_{i-1}, n, n_{i+1}, \ldots, n_m)}{\sum_j p(n_1, \ldots, n_{i-1}, j, n_{i+1}, \ldots, n_m)} \\ &= \frac{P_i(n)}{\sum_j P_i(j)} \end{aligned}$$

where the preceding sum is over all $j = 0, \ldots, r - \sum_{k\neq i} n_k$. In other words, the conditional distribution of N_i given the values of N_j, $j \neq i$, is the same as the conditional distribution of a random variable having mass function P_i given that its value is less than or equal to $r - \sum_{j\neq i} N_j$.

Thus, we may generate the values of a Markov chain whose limiting probability mass function is $p(n_1, \ldots, n_m)$ as follows:

1. Let $(n_1, \ldots, n_m)$ be arbitrary nonnegative integers satisfying $\sum_i n_i \leq r$.
2. Generate U and let $I = \text{Int}(mU + 1)$.
3. If $I = i$, let X_i have mass function P_i and generate a random variable N whose distribution is the conditional distribution of X_i given that $X_i \leq r - \sum_{j \neq i} n_j$.
4. Let $n_i = N$ and go to 2.

The successive values of $(n_1, \ldots, n_m)$ constitute the sequence of states of a Markov chain with the limiting distribution p. All quantities of interest concerning p can be estimated from this sequence. For instance, the average of the values of the jth coordinate of these vectors will converge to the mean number of individuals at station j, the proportion of vectors whose jth coordinate is less than k will converge to the limiting probability that the number of individuals at station j is less than k, and so on. □

Example 12d Let $X_i, i = 1, \ldots, n$, be independent random variables with X_i having an exponential distribution with rate $\lambda_i, i = 1, \ldots, n$. Let $S = \sum_{i=1}^n X_i$ and suppose we want to generate the random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ conditional on the event that $S > c$ for some large positive constant c. That is, we want to generate the value of a random vector whose density function is given by

$$f(x_1, \ldots, x_n) = \frac{1}{P\{S > c\}} \prod_{i=1}^n \lambda_i e^{-\lambda_i x_i}, \quad \text{if} \sum_{i=1}^n x_i > c$$

This is easily accomplished by starting with an initial vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ satisfying $x_i > 0, i = 1, \ldots, n$, and $\sum_{i=1}^n x_i > c$. Then generate a random number U and set $I = \text{Int}(nU + 1)$. Suppose that $I = i$. Now, we want to generate an exponential random variable X with rate λ_i conditioned on the event that $X + \sum_{j \neq i} x_j > c$. That is, we want to generate the value of X conditional on the event that it exceeds $c - \sum_{j \neq i} x_j$. Hence, using the fact that an exponential conditioned to be greater than a positive constant is distributed as the constant plus the exponential, we see that we should generate an exponential random variable Y with rate λ_i (say, let $Y = -1/\lambda_i \log U$), and set

$$X = Y + \left(c - \sum_{j \neq i} x_j\right)^+$$

where b^+ is equal to b when $b > 0$ and is 0 otherwise. The value of x_i should then be reset to equal X and a new iteration of the algorithm begun. □

Suppose now that we interested in estimating

$$\alpha = P\{h(\boldsymbol{X}) > a\}$$

where $\boldsymbol{X} = (X_1, \ldots, X_n)$ is a random vector, h is an arbitrary function of $\boldsymbol{X}$, and α is very small. Because a generated value of $h(\boldsymbol{X})$ will almost always be less than a, it would take a huge amount of time to obtain an estimator whose error is small relative to α if we use a straightforward Gibbs sampler approach to generate a sequence of random vectors whose distribution converges to that of $\boldsymbol{X}$. Consider, however, the following approach.

To begin, note that for values $-\infty = a_0 < a_1 < a_2 < \cdots < a_k = a$,

$$\alpha = \prod_{i=1}^{k} P\{h(\boldsymbol{X}) > a_i | h(\boldsymbol{X}) > a_{i-1}\}$$

Thus, we can obtain an estimator of α by taking the product of estimators of the quantities $P\{h(\boldsymbol{X}) > a_i | h(\boldsymbol{X}) > a_{i-1}\}$, for $i = 1, \ldots, k$. For this to be efficient, the values $a_i, i = 1, \ldots, k$, should be chosen so that $P\{h(\boldsymbol{X}) > a_i | h(\boldsymbol{X}) > a_{i-1}\}$ are all of moderate size.

To estimate $P\{h(\boldsymbol{X}) > a_i | h(\boldsymbol{X}) > a_{i-1}\}$, we make use of the Gibbs sampler as follows.

```
1. Set $J = N = 0$.
2. Choose a vector $\boldsymbol{x}$ such that $h(\boldsymbol{x}) > a_{i-1}$.
3. Generate a random number $U$ and set $I = \text{Int}(nU) + 1$.
4. If $I = k$, generate $X$ having the conditional distribution of $X_k$ given that
   $X_j = x_j, j \neq k$.
5. If $h(x_1, \ldots, x_{k-1}, X, x_{k+1}, \ldots, x_n) \leq a_{i-1}$, return to 4.
6. $N = N + 1, x_k = X$.
7. If $h(x_1, \ldots, x_n) > a_i$ then $J = J + 1$.
8. Go to 3.
```

The ratio of the final value of J to that of N is the estimator of $P\{h(\boldsymbol{X}) > a_i | h(\boldsymbol{X}) > a_{i-1}\}$.

Example 12e Suppose in the queueing network model of Example 12d that the service times at server i are exponential with rate $\mu_i, i = 1, \ldots, m + 1$, and that when a customer completes service at server i then, independent of all else, that customer then moves over to join the queue (or enter service if the server is free) at server j with probability P_{ij}, where $\sum_{j=1}^{m+1} P_{ij} = 1$. It can then be shown that the limiting probability mass function of the number of customers at servers $1, \ldots, m$ is given, for $\sum_{j=1}^{m} n_j \leq r$, by

$$p(n_1, \ldots, n_m) = C \prod_{j=1}^{m} \left(\frac{\pi_j \mu_{m+1}}{\pi_{m+1} \mu_j} \right)^{n_j}$$

where π_j, $j = 1, \ldots, m+1$, are the stationary probabilities of the Markov chain with transition probabilities P_{ij}. That is, they are the unique solution of

$$\pi_j = \sum_{i=1}^{m+1} \pi_i P_{ij}$$
$$\sum_{j=1}^{m+1} \pi_j = 1$$

If we renumber the servers so that $\max(\pi_j/\mu_j) = \pi_{m+1}/\mu_{m+1}$, then letting $a_j = \pi_j \mu_{m+1}/\pi_{m+1}\mu_j$, we have that for $\sum_{j=1}^{m} n_j \leq r$,

$$p(n_1, \ldots, n_m) = C \prod_{j=1}^{m} (a_j)^{n_j}$$

where $0 \leq a_j \leq 1$. It easily follows from this that the conditional distribution of the number of customers at server i, given the numbers n_j, $j \neq i$, at the other $m-1$ servers, is distributed as the conditional distribution of -1 plus a geometric random variable with parameter $1 - a_i$, given that the geometric is less than or equal to $r + 1 - \sum_{j \neq i} n_j$.

In the case where the π_j and μ_j are both constant for all j, the conditional distribution of the number of customers at server i, given the numbers n_j, $j \neq i$, at the other servers excluding server $m+1$, is the discrete uniform distribution on $0, 1, \ldots, r - \sum_{j \neq i} n_j$. Suppose this is the case and that $m = 20, r = 100$, and that we are interested in estimating the limiting probability that the number of customers at server 1—call it X_1—is greater than 18. Letting $t_0 = -1, t_1 = 5, t_2 = 9, t_3 = 12, t_4 = 15, t_5 = 17, t_6 = 18$, we can use the Gibbs sampler to successively estimate the quantities $P\{X_1 > t_i | X_1 > t_{i-1}\}, i = 1, 2, 3, 4, 5, 6$. We would estimate, say $P\{X_1 > 17 | X_1 > 15\}$, by starting with a vector $n_1, \ldots, n_{20}$ for which $n_1 > 15$ and $s = \sum_{i=1}^{20} n_i \leq 100$. We then generate a random number U and let $I = \text{Int}(20U + 1)$. A second random number V is now generated. If $I = 1$, then n_1 is reset to

$$n_1 = \text{Int}((85 - s + n_1)V) + 16$$

If $I \neq 1$, then n_1 is reset to

$$n_1 = \text{Int}((101 - s + n_1)V)$$

The next iteration of the algorithm then begins; the fraction of iterations for which $n_1 > 17$ is the estimate of $P\{X_1 > 17 | X_1 > 15\}$. □

The idea of writing a small probability as the product of more moderately sized conditional probabilities and then estimating each of the conditional probabilities in turn does not require that the Gibbs sampler be employed. Another variant of the Hastings–Metropolis algorithm might be more appropriate. We illustrate by an example that was previously treated, in Example 9v, by using importance sampling.

Example 12f Suppose that we are interested in estimating the number of permutations $x = (x_1, \ldots, x_n)$ for which $t(x) > a$, and where $t(x) = \sum_{j=1}^{n} jx_j$ and where a is such that this number of permutations is very small in comparison to $n!$. If we let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be equally likely to be any of the $n!$ permutations and set

$$\alpha = P\{T(\boldsymbol{X}) > a\}$$

then α is small and the quantity of interest is $\alpha n!$. Letting $0 = a_0 < a_1 < \cdots < a_k = a$, we have that

$$\alpha = \prod_{i=1}^{k} P\{T(\boldsymbol{X}) > a_i | T(\boldsymbol{X}) > a_{i-1}\}$$

To estimate $P\{T(\boldsymbol{X}) > a_i | T(\boldsymbol{X}) > a_{i-1}\}$ we use the Hastings–Metropolis algorithm as in Examples 12a or 12b to generate a Markov chain whose limiting distribution is

$$\pi(x) = \frac{1}{N_{i-1}}, \quad \text{if } T(x) > a_{i-1}$$

where N_{i-1} is the number of permutations x such that $T(x) > a_{i-1}$. The proportion of the generated states x of this Markov chain that have $T(x) > a_i$ is the estimate of $P\{T(\boldsymbol{X}) > a_i | T(\boldsymbol{X}) > a_{i-1}\}$. □

In many applications it is relatively easy to recognize the form of the conditional distributions needed in the Gibbs sampler.

Example 12g Suppose that for some nonnegative function $h(y, z)$ the joint density of the nonnegative random variables X, Y, and Z is

$$f(x, y, z) = Cx^{y-1}(1-x)^{zy}h(y, z), \quad \text{for } 0 < x < 0.5$$

Then the conditional density of X given that $Y = y$ and $Z = z$ is

$$f(x|y, z) = \frac{f(x, y, z)}{f_{Y,Z}(y, z)}$$

Since y and z are fixed and x is the argument of this conditional density, we can write the preceding as

$$f(x|y, z) = C_1 f(x, y, z)$$

where C_1 does not depend on x. Hence, we have that

$$f(x|y, z) = C_2 x^{y-1}(1-x)^{zy}, \quad 0 < x < 0.5$$

where C_2 does not depend on x. But we can recognize this as the conditional density of a beta random variable with parameters y and $zy + 1$ that is conditioned to be in the interval (0, 0.5). □

Rather than always choosing a random coordinate to update on, the Gibbs sampler can also consider the coordinates in sequence. That is, on the first iteration we could set $I = 1$, then set $I = 2$ on the next iteration, then $I = 3$, and so on until the nth iteration, where $I = n$. On the next iteration, we start over. We illustrate this with our next example, which is concerned with modeling the numbers of home runs hit by two of the best hitters in baseball.

Example 12h Let $N_1(t)$ denote the number of home runs hit in the first $100t$ percent of a baseball season, $0 \le t \le 1$, by the baseball player AB; similarly, let $N_2(t)$ be the number hit by CD.

Suppose that there are random variables W_1 and W_2 such that given that $W_1 = w_1$ and $W_2 = w_2$, $\{N_1(t), 0 \le t \le 1\}$ and $\{N_2(t), 0 \le t \le 1\}$ are independent Poisson processes with respective rates w_1 and w_2. Furthermore, suppose that W_1 and W_2 are independent exponential random variables with rate Y, which is itself a random variable that is uniformly distributed between 0.02 and 0.10. In other words, the assumption is that the players hit home runs in accordance with Poisson processes whose rates are random variables from a distribution that is defined in terms of a parameter that is itself a random variable with a specified distribution.

Suppose that AB has hit 25 and CD 18 home runs in the first half of the season. Give a method for estimating the mean number they each hit in the full season.

Solution Summing up the model, there are random variables Y, W_1, W_2 such that:

1. Y is uniform on (0.02, 0.10).
2. Given that $Y = y$, W_1 and W_2 are independent and identically distributed exponential random variables with rate y.
3. Given that $W_1 = w_1$ and $W_2 = w_2$, $\{N_1(t)\}$ and $\{N_2(t)\}$ are independent Poisson processes with rates w_1 and w_2.

To find $E[N_1(1)|N_1(0.5) = 25, N_2(0.5) = 18]$, start by conditioning on W_1.

$$E[N_1(1)|N_1(0.5) = 25, N_2(0.5) = 18, W_1] = 25 + 0.5W_1$$

Taking the conditional expectation, given that $N_1(0.5) = 25$ and $N_2(0.5) = 18$, of the preceding yields that

$$\begin{aligned} &E[N_1(1)|N_1(0.5) = 25, N_2(0.5) = 18] \\ &\quad = 25 + 0.5E[W_1|N_1(0.5) = 25, N_2(0.5) = 18] \end{aligned}$$

Similarly,

$$\begin{aligned} &E[N_2(1)|N_1(0.5) = 25, N_2(0.5) = 18] \\ &\quad = 18 + 0.5E[W_2|N_1(0.5) = 25, N_2(0.5) = 18] \end{aligned}$$

We can now estimate these conditional expectations by using the Gibbs sampler. To begin, note the joint distribution: For $0.02 < y < 0.10$, $w_1 > 0$, $w_2 > 0$,

$$\begin{aligned} &f(y, w_1, w_2, N_1(0.5) = 25, N_2(0.5) = 18) \\ &\quad = Cy^2 e^{-(w_1+w_2)y} e^{-(w_1+w_2)/2} (w_1)^{25} (w_2)^{18} \end{aligned}$$

where C does not depend on any of y, w_1, w_2. Hence, for $0.02 < y < 0.10$,

$$f(y|w_1, w_2, N_1 = 25, N_2 = 18) = C_1 y^2 e^{-(w_1+w_2)y}$$

which shows that the conditional distribution of Y given $w_1, w_2, N_1 = 25$, $N_2 = 18$, is that of a gamma random variable with parameters 3 and $w_1 + w_2$ that is conditioned to be between 0.02 and 0.10. Also,

$$f(w_1|y, w_2, N_1(0.5) = 25, N_2(0.5) = 18) = C_2 e^{-(y+1/2)w_1} (w_1)^{25}$$

from which we can conclude that the conditional distribution of W_1 given $y, w_2, N_1 = 25, N_2 = 18$ is gamma with parameters 26 and $y + \frac{1}{2}$. Similarly, the conditional distribution of W_2 given $y, w_1, N_1 = 25, N_2 = 18$, is gamma with parameters 19 and $y + \frac{1}{2}$.

Hence, starting with values y, w_1, w_2, where $.02 < y < 0.10$, and $w_i > 0$, the Gibbs sampler is as follows.

1. Generate the value of a gamma random variable with parameters 3 and $w_1 + w_2$ that is conditioned to be between or 0.02 and 0.10, and let it be the new value of y.
2. Generate the value of a gamma random variable with parameters 26 and $y + \frac{1}{2}$, and let it be the new value of w_1.
3. Generate the value of a gamma random variable with parameters 19 and $y + \frac{1}{2}$, and let it be the new value of w_2.
4. Return to Step 1.

The average of the values of w_1 is our estimate of $E[W_1|N_1(0.5) = 25, N_2(0.5) = 18]$, and the average of the values of w_2 is our estimate of $E[W_2|N_1(0.5) = 25, N_2(0.5) = 18]$. One-half of the former plus 25 is our estimate of the mean number of home runs that AB will hit over the year, and one-half of the latter plus 18 is our estimate of the mean number that CD will hit.

It should be noted that the numbers of home runs hit by the two players are dependent, with their dependence caused by their common dependence on the value of the random variable Y. That is, the value of Y (which might relate to such quantities as the average degree of liveliness of the baseballs used that season or the average weather conditions for the year) affects the distribution of the mean number of home runs that each player will hit in the year. Thus, information about the number of home runs hit by one of the players yields probabilistic information about the value of Y that affects the distribution of the number of home runs of the other player. This type of model, where there is a common random variable (Y in this case) that affects the distributions of the conditional parameters of the random variables of interest, is known as an *hierarchical Bayes* model. □

When applying the Gibbs sampler, it is not necessary to condition on all but one of the variables. If it is possible to generate from joint conditional distributions, then we may utilize them. For instance, suppose $n = 3$ and that we can generate from the conditional distribution of any two of them given the third. Then, at each iteration we could generate a random number U, set $I = \text{Int}(3U+1)$, and generate from the joint distribution of $X_j, X_k, j, k \neq I$, given the present value of X_I.

Example 12i Let $X_i, i = 1, 2, 3, 4, 5$, be independent exponential random variables, with X_i having mean i, and suppose we are interested in using simulation to estimate

$$\beta = P\left\{\prod_{i=1}^{5} X_i > 120 \middle| \sum_{i=1}^{5} X_i = 15\right\}$$

We can accomplish this by using the Gibbs sampler via a random choice of two of the coordinates. To begin, suppose that X and Y are independent exponentials with respective rates λ and μ, where $\mu < \lambda$, and let us find the conditional distribution of X given that $X + Y = a$, as follows.

$$\begin{aligned} f_{X|X+Y}(x|a) &= C_1 f_{X,Y}(x, a - x), \quad 0 < x < a \\ &= C_2 e^{-\lambda x} e^{-\mu(a-x)}, 0 < x < a \\ &= C_3 e^{-(\lambda-\mu)x}, 0 < x < a \end{aligned}$$

which shows that the conditional distribution is that of an exponential with rate $\lambda - \mu$ that is conditioned to be less than a.

Using this result, we can estimate β by letting the initial state $(x_1, x_2, x_3, x_4, x_5)$ be any five positive numbers that sum to 15. Now randomly choose two elements from the set 1, 2, 3, 4, 5; say $I = 2$ and $J = 5$ are chosen. Then the conditional distribution of X_2, X_5 given the other values is the conditional distribution of two independent exponentials with means 2 and 5, given that their sum is $15 - x_1 - x_3 - x_4$. But, by the preceding, the values of X_2 and X_5 can be obtained by generating the value of an exponential with rate $\frac{1}{2} - \frac{1}{5} = \frac{3}{10}$ that is conditioned to be less than $15 - x_1 - x_3 - x_4$, then setting x_2 equal to that value and resetting x_5 to make $\sum_{i=1}^{5} x_i = 15$. This process should be continually repeated, and the proportion of state vectors $\mathbf{x}$ having $\prod_{i=1}^{5} x_i > 120$ is the estimate of β. □

Example 12j Suppose that n independent trials are performed; each of which results in one of the outcomes $1, 2, \ldots, r$, with respective probabilities $p_1, p_2, \ldots, p_r, \sum_{i=1}^{r} p_i = 1$, and let X_i denote the number of trials that result in outcome i. The random variables $X_1, \ldots, X_r$, whose joint distribution is called the multinomial distribution, were introduced in Example 12g where it was shown how they can be simulated. Now suppose $n > r$, and that we want to simulate

$X_1, \ldots, X_r$ conditional on the event that they are all positive. That is, we want to simulate the result of the trials conditional on the event that each outcome occurs at least once. How can this be efficiently accomplished when this conditioning event has a very small probability?

Solution To begin, it should be noted that it would be wrong to suppose that we could just generate the result of $n - r$ of these trials, and then let X_i equal 1 plus the number of these $n - r$ trials that result in outcome i. (That is, attempting to put aside the r trials in which all outcomes occur once, and then simulating the remaining $n - r$ trials does not work.) To see why, let $n = 4$ and $r = 2$. Then, under the putting aside method, the probability that exactly 2 of the trials would result in outcome 1 is $2p(1 - p)$, where $p = p_1$. However, for the multinomial random variables X_1, X_2

$$\begin{aligned} P\{X_1 = 2|X_1 > 0, X_2 > 0\} &= \frac{P\{X_1 = 2\}}{P\{X_1 > 0, X_2 > 0\}} \\ &= \frac{P\{X_1 = 2\}}{1 - P\{X_1 = 4\} - P\{X_2 = 4\}} \\ &= \frac{\binom{4}{2} p^2(1-p)^2}{1 - p^4 - (1-p)^4} \end{aligned}$$

As the preceding is not equal to $2p(1 - p)$(try$p = 1/2$), the method does not work.

We can use the Gibbs sampler to generate a Markov chain having the appropriate limiting probabilities. Let the initial state be any arbitrary vector of r positive integers whose sum is n, and let the states change in the following manner. Whenever the state is $x_1, \ldots, x_r$, generate the next state by first randomly choosing two of the indices from $1, \ldots, r$. If i and j are chosen, let $s = x_i + x_j$, and simulate X_i and X_j from their conditional distribution given that $X_k = x_k, k \neq i, j$. Because conditional on $X_k = x_k, k \neq i, j$ there are a total of s trials that result in either outcome i or j, it follows that the number of these trials that result in outcome i is distributed as a binomial random variable with parameters $(s, \frac{p_i}{p_i + p_j})$ that is conditioned to be one of the values $1, \ldots, s - 1$. Consequently, the discrete inverse transform method can be used to simulate such a random variable; if its value is v, then the next state is the same as the previous one with the exception that the new values of x_i and x_j are v and $s - v$. Continuing on in this manner results in a sequence of states whose limiting distribution is that of the multinomial conditional on the event that all outcomes occur at least once. □

Remarks

1. The same argument can be used to verify that we obtain the appropriate limiting mass function when we consider the coordinates in sequence and

apply the Gibbs sampler (as in Example 12i), or when we use it via conditioning on less than all but one of the values (as in Example 12j). These results are proven by noticing that if one chooses the initial state according to the mass function f, then, in either case, the next state also has mass function f. But this shows that f satisfies the Equations (12.2), implying by uniqueness that f is the limiting mass function.

2. Suppose you are using the Gibbs sampler to estimate $E[X_i]$ in a situation where the conditional means $E[X_i|X_j, j \neq i]$ are easily computed. Then, rather than using the average of the successive values of X_i as the estimator, it is usually better to use the average of the conditional expectations. That is, if the present state is $\boldsymbol{x}$, then take $E[X_i|X_j = x_j, j \neq i]$ rather than x_i as the estimate from that iteration. Similarly, if you are trying to estimate $P\{X_i = x\}$, and $P\{X_i = x|X_j, j \neq i\}$ is easily computed, then the average of these quantities is usually a better estimator than is the proportion of time in which the ith component of the state vector equals x.
3. The Gibbs sampler shows that knowledge of all the conditional distributions of X_i given the values of the other X_j, $j \neq i$, determines the joint distribution of $\boldsymbol{X}$. □

12.4 Continuous time Markov Chains and a Queueing Loss Model

We often are interested in a process $\{X(t), t \geq 0\}$ that evolves continuously over time. Interpreting $X(t)$ as the state of the process at time t, the process is said to be a *continuous time Markov chain* having stationary transition probabilities if the set of possible states is either finite or countably infinite, and the process satisfies the following properties:
Given that the current state is i, then

(a) the time until the process makes a transition into another state is an exponential random variable with rate, say, v_i;
(b) when a transition out of state i occurs then, independent of what has previously occurred, including how long it has taken to make the transition from state i, the next state entered will be j with probability $P_{i,j}$.

Thus, while the sequence of states of a continuous time Markov chain constitutes a discrete time Markov chain with transition probabilities $P_{i,j}$, the times between transitions are exponentially distributed with rates depending on the current state. Let us suppose that the chain has a finite number of states, which in our general discussion we label as $1, \ldots, N$.

Let $P(i)$ denote the long run proportion of time that the chain is in state i. (Assuming that the discrete time Markov chain composed of the sequence of states is irreducible, these long run proportions will exist and will not depend on the initial state of the process. Also, because the time spent in a state has a continuous exponential distribution, there is no analog to a periodic discrete time chain and so the long run proportions are always also limiting probabilities.) If we let

$$\lambda(i, j) = v_i P_{i,j}$$

then because v_i is the rate at which the chain when in state i makes a transition out of that state, and $P_{i,j}$ is the probability that such a transition is into state j, it follows that $\lambda_{(i,j)}$ is the rate when in state i that the chain makes a transition into state j. The continuous time Markov chain is said to be *time reversible* if

$$P(i)\lambda(i, j) = P(j)\lambda(j, i), \quad \text{for all } i, j$$

Thus, the continuous time Markov chain will be time reversible if the rate of transitions from i to j is equal to rate of transitions from j to i, for all states i and j. Moreover, as in the case of a discrete time Markov chain, if one can find probabilities $P(i), i = 1, \ldots, N$ that satisfy the preceding *time reversibility* equations, then the chain is time reversible and the $P(i)$ are the limiting (also known as *stationary*) probabilities.

Let us now consider a queueing system in which customers arrive according to a Poisson process with rate λ. Suppose that each customer is of one of the types $1, \ldots, r$, and that each new arrival is, independent of the past, a type i customer with probability p_i, $\sum_{i=1}^{r} p_i = 1$. Suppose that if a type i customer is allowed to enter the system, then the time it spends before departing is an exponential random variable with rate μ_i, $i = 1, \ldots, r$. Further suppose that the decision as to whether or not to allow a type i customer to enter depends on the set of customers currently in the system. More specifically, say that the current state of the system is $(n_1, \ldots, n_r)$ if there are currently n_i type i customers in the system, for each $i = 1, \ldots, r$, and suppose that there is a specified set of states $\mathcal{A}$ such that a customer would not be allowed into the system if that would result in a system state that is not in $\mathcal{A}$. That is, if the current state is $\mathbf{n} = (n_1, \ldots, n_r) \in \mathcal{A}$ when a type i customer arrives, then that customer would be allowed to enter the system if $\mathbf{n} + \mathbf{e_i} \in \mathcal{A}$, and would not be allowed to enter if $\mathbf{n} + \mathbf{e_i} \notin \mathcal{A}$, where $\mathbf{e_i} = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 being in position i. Suppose further that $\mathcal{A}$ is such that $\mathbf{n} + \mathbf{e_i} \in \mathcal{A}$ implies that $\mathbf{n} \in \mathcal{A}$.

For an example of the preceding, suppose the system is a hospital and that the arrivals are patients. Suppose that the hospital provides m types of services and that a type i patient requires $r_i(j) \geq 0$ units of service type j. If we further suppose that the hospital's capacity for providing type j service is $c_j \geq 0$, it follows that the hospital can simultaneously accommodate n_1 type 1 patients, n_2 type 2 patients, $\ldots$, and n_r type r patients if

$$\sum_{i=1}^{r} n_i r_i(j) \leq c_j, \quad j = 1, \ldots, m$$

and so

$$\mathcal{A} = \{\mathbf{n} : \sum_{i=1}^{r} n_i r_i(j) \le c_j, j = 1, \ldots, m\}$$

We now show that the continuous time Markov chain with states $\mathbf{n} \in \mathcal{A}$ is time reversible. To do so, suppose that $\mathbf{n} = (n_1, \ldots, n_r) \in \mathcal{A}$, with $n_i > 0$. Note that when in state $\mathbf{n}$ the process will go to state $\mathbf{n} - \mathbf{e_i}$ if a type i customer departs; as there are n_i type i customers in the system this will occur at rate $n_i \mu_j$. Hence, if $P(\mathbf{n})$ is the proportion of time that the state is $\mathbf{n}$,we see that

$$\text{rate at which the process goes from state } \mathbf{n} \text{ to state } \mathbf{n} - \mathbf{e_i} = P(\mathbf{n}) n_i \mu_i$$

In addition, when in state $\mathbf{n} - \mathbf{e_i}$ the rate at which the process goes to state $\mathbf{n}$ is the arrival rate of a type i customer, namely λp_i. Consequently, with $\lambda_i \equiv \lambda p_i$,

$$\text{rate at which the process goes from state } \mathbf{n} - \mathbf{e_i} \text{ to state } \mathbf{n} = P(\mathbf{n} - \mathbf{e_i}) \lambda_i$$

Thus the time reversibility equations are

$$P(\mathbf{n}) n_i \mu_i = P(\mathbf{n} - \mathbf{e_i}) \lambda_i$$

Solving the preceding for $P(\mathbf{n})$ and then iterating this solution n_i times yields that

$$\begin{aligned}
P(\mathbf{n}) &= \frac{\lambda_i/\mu_i}{n_i} P(\mathbf{n} - \mathbf{e_i}) \\
&= \frac{\lambda_i/\mu_i}{n_i} \frac{\lambda_i/\mu_i}{(n_i - 1)} P(\mathbf{n} - \mathbf{e_i} - \mathbf{e_i}) \\
&= \frac{(\lambda_i/\mu_i)^2}{n_i(n_i - 1)} P(\mathbf{n} - \mathbf{e_i} - \mathbf{e_i}) \\
&= \ldots \\
&= \ldots \\
&= \ldots \\
&= \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!} P(n_1, \ldots, n_{i-1}, 0, n_{i+1}, \ldots, n_r)
\end{aligned}$$

Doing the same with the other coordinates of the vector $\mathbf{n}$ shows that the time reversibility equations yield that

$$P(\mathbf{n}) = P(\mathbf{0}) \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}$$

To determine $P(\mathbf{0}) = P(0, \ldots, 0)$, we sum the preceding over all vectors $\mathbf{n} \in \mathcal{A}$, which yields that

$$1 = P(\mathbf{0}) \sum_{\mathbf{n} \in \mathcal{A}} \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}$$

Hence, the time reversibility equations imply that

$$P(\mathbf{n}) = \frac{\prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}}{\sum_{\mathbf{n}\in\mathcal{A}} \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}} = C \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}, \quad \mathbf{n} \in \mathcal{A} \tag{12.4}$$

where $C = \frac{1}{\sum_{\mathbf{n}\in\mathcal{A}} \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}}$. Because the preceding formulas for $P(\mathbf{n})$ are easily shown to satisfy the time reversibility equations, we can thus conclude that the chain is time reversible with stationary probabilities given by (12.4). It is, however, difficult to directly use the preceding formula because it would not normally be computationally possible to compute C. However, we can use the Markov chain monte carlo method to great effect, as we now show.

To start, note that if $X_1, \ldots, X_r$ are independent Poisson random variables, with X_i having mean λ_i/μ_i, then the stationary distribution given by (12.4) is the conditional distribution of $\mathbf{X} = (X_1, \ldots, X_r)$ given that $\mathbf{X} \in \mathcal{A}$. This is so, because for $\mathbf{n} = (n_1, \ldots, n_r) \in \mathcal{A}$

$$\begin{aligned} P(X_i = n_i, i = 1, \ldots, r|\mathbf{X} \in \mathcal{A}) &= \frac{\prod_{i=1}^{r} P(X_i = n_i)}{P(\mathbf{X} \in \mathcal{A})} \\ &= \frac{\prod_{i=1}^{r} e^{-\lambda_i/\mu_i} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!}}{P(\mathbf{X} \in \mathcal{A})} \\ &= K \prod_{i=1}^{r} \frac{(\lambda_i/\mu_i)^{n_i}}{n_i!} \end{aligned}$$

where $K = e^{-\sum_i \lambda_i/\mu_i}/P(\mathbf{X} \in \mathcal{A})$is a constant that does not depend on $\mathbf{n}$. Because the sums, over all $\mathbf{n} \in \mathcal{A}$, of both the preceding and the mass function given by (12.4) equal 1, we see that $K = C$, and so the stationary distribution of the continuous time Markov chain is the conditional distribution of $\mathbf{X}$ given that $\mathbf{X} \in \mathcal{A}$. Now, the conditional distribution of X_i given $X_j = n_j, j \neq i, \mathbf{X} \in \mathcal{A}$, is that of a Poisson random variable X_i with mean λ_i/μ_i that is conditioned to be such that $(n_1, \ldots, n_{i-1}, X_i, n_{i+1}, \ldots, n_r) \in \mathcal{A}$. Because $\mathbf{n} + \mathbf{e_i} \in \mathcal{A}$ implies that $\mathbf{n} \in \mathcal{A}$, the preceding conditional distribution will be the distribution of a Poisson random variable X_i with mean λ_i/μ_i that is conditioned to be less than or equal to $v \equiv \max\{k : (n_1, \ldots, n_{i-1}, k, n_{i+1}, \ldots, n_r) \in \mathcal{A}\}$. As such a random variable is easily generated, say by the discrete inverse transform technique, we see that the Gibb's sampler can be effectively employed to generate a Markov chain whose limiting distribution is the stationary distribution of the queueing model.

12.5 Simulated Annealing

Let $\mathcal{A}$ be a finite set of vectors and let $V(\mathbf{x})$ be a nonnegative function defined on $\mathbf{x} \in \mathcal{A}$, and suppose that we are interested in finding its maximal value and at least one argument at which the maximal value is attained. That is, letting

$$V^* = \max_{x \in \mathcal{A}} V(\mathbf{x})$$

and

$$\mathcal{M} = \{\mathbf{x} \in \mathcal{A} : V(\mathbf{x}) = V^*\}$$

we are interested in finding V^* as well as an element in $\mathcal{M}$. We will now show how this can be accomplished by using the methods of this chapter.

To begin, let $\lambda > 0$ and consider the following probability mass function on the set of values in $\mathcal{A}$:

$$p_\lambda(\mathbf{x}) = \frac{e^{\lambda V(\mathbf{x})}}{\sum_{\mathbf{x} \in \mathcal{A}} e^{\lambda V(\mathbf{x})}}$$

By multiplying the numerator and denominator of the preceding by $e^{-\lambda V^*}$, and letting $|\mathcal{M}|$ denote the number of elements in $\mathcal{M}$, we see that

$$p_\lambda(\mathbf{x}) = \frac{e^{\lambda(V(\mathbf{x}) - V^*)}}{|\mathcal{M}| + \sum_{\mathbf{x} \notin \mathcal{M}} e^{\lambda(V(\mathbf{x}) - V^*)}}$$

However, since $V(\mathbf{x}) - V^* < 0$ for $\mathbf{x} \notin \mathcal{M}$, we obtain that as $\lambda \to \infty$,

$$p_\lambda(\mathbf{x}) \to \frac{\delta(\mathbf{x}, \mathcal{M})}{|\mathcal{M}|}$$

where $\delta(\mathbf{x}, \mathcal{M}) = 1$ if $\mathbf{x} \in \mathcal{M}$ and is 0 otherwise.

Hence, if we let λ be large and generate a Markov chain whose limiting distribution is $p_\lambda(\mathbf{x})$, then most of the mass of this limiting distribution will be concentrated on points in $\mathcal{M}$. An approach that is often useful in defining such a chain is to introduce the concept of neighboring vectors and then use a Hastings–Metropolis algorithm. For instance, we could say that the two vectors $\mathbf{x} \in \mathcal{A}$ and $\mathbf{y} \in \mathcal{A}$ are neighbors if they differ in only a single coordinate or if one can be obtained from the other by interchanging two of its components. We could then let the target next state from $\mathbf{x}$ be equally likely to be any of its neighbors, and if the neighbor $\mathbf{y}$ is chosen, then the next state becomes $\mathbf{y}$ with probability

$$\min\left\{1, \frac{e^{\lambda V(\mathbf{y})}/|N(\mathbf{y})|}{e^{\lambda V(\mathbf{x})}/|N(\mathbf{x})|}\right\}$$

or remains $\mathbf{x}$ otherwise, where $|N(\mathbf{z})|$ is the number of neighbors of $\mathbf{z}$. If each vector has the same number of neighbors (and if not already so, this can almost always be arranged by increasing the state space and letting the V value of any new state equal 0), then when the state is $\mathbf{x}$, one of its neighbors, say $\mathbf{y}$, is randomly chosen; if $V(\mathbf{y}) \geq V(\mathbf{x})$, then the chain moves to state $\mathbf{y}$, and if $V(\mathbf{y}) < V(\mathbf{x})$, then the chain moves to state $\mathbf{y}$ with probability $\exp\{\lambda(V(\mathbf{y}) - V(\mathbf{x}))\}$ or remains in state $\mathbf{x}$ otherwise.

One weakness with the preceding algorithm is that because λ was chosen to be large, if the chain enters a state $\mathbf{x}$ whose V value is greater than that of each of its neighbors, then it might take a long time for the chain to move to a different state. That is, whereas a large value of λ is needed for the limiting distribution to put most of its weight on points in $\mathcal{M}$, such a value typically requires a very large number of transitions before the limiting distribution is approached. A second weakness is that since there are only a finite number of possible values of $\mathbf{x}$, the whole concept of convergence seems meaningless since we could always, in theory, just try each of the possible values and so obtain convergence in a finite number of steps. Thus, rather than considering the preceding from a strictly mathematical point of view, it makes more sense to regard it as a heuristic approach, and in doing so it has been found to be useful to allow the value of λ to change with time.

A popular variation of the preceding, known as *simulated annealing*, operates as follows. If the nth state of the Markov chain is $\mathbf{x}$, then a neighboring value is randomly selected. If it is $\mathbf{y}$, then the next state is either $\mathbf{y}$ with probability

$$\min\left\{1, \frac{\exp\{\lambda_n V(\mathbf{y})\}/|N(\mathbf{y})|}{\exp\{\lambda_n V(\mathbf{x})\}/|N(\mathbf{x})|}\right\}$$

or it remains $\mathbf{x}$, where $\lambda_n, n \geq 1$, is a prescribed set of values that start out small (thus resulting in a large number of changes in state) and then grow.

A computationally useful choice of λ_n (and a choice that mathematically results in convergence) is to let $\lambda_n = C\log(1+n)$, where $C > 0$ is any fixed positive constant (see Besag et al., 1995; Diaconis and Holmes 1995). If we then generate m successive states $X_1, \ldots, X_m$, we can then estimate V^* by $\max_{i=1\ldots,m} V(X_i)$, and if the maximum occurs at X_{i^*} then this is taken as an estimated point in $\mathcal{M}$.

Example 12k The Traveling Salesman Problem One version of the traveling salesman problem is for the salesman to start at city 0 and then sequentially visit all of the cities $1, \ldots, r$. A possible choice is then a permutation $x_1, \ldots, x_r$ of $1, \ldots, r$ with the interpretation that from 0 the salesman goes to city x_1, then to x_2, and so on. If we suppose that a nonnegative reward $v(i, j)$ is earned whenever the salesman goes directly from city i to city j, then the return of the choice $\mathbf{x} = (x_1, \ldots, x_r)$ is

$$V(\mathbf{x}) = \sum_{i=1}^{r} v(x_{i-1}, x_i) \quad \text{where } x_0 = 0$$

By letting two permutations be neighbors if one results from an interchange of two of the coordinates of the other, we can use simulated annealing to approximate the best path. Namely, start with any permutation $\mathbf{x}$ and let $X_0 = \mathbf{x}$. Now, once the nth state (that is, permutation) has been determined, $n \geq 0$, then generate one of its neighbors at random [by choosing I, J equally likely to be any of the $\binom{r}{2}$

values $i \neq j, i, j = 1, \ldots, r$ and then interchanging the values of the Ith and Jth elements of X_n]. Let the generated neighbor be $\mathbf{y}$. Then if $V(\mathbf{y}) \geq V(X_n)$, set $X_{n+1} = \mathbf{y}$. Otherwise, set $X_{n+1} = \mathbf{y}$ with probability $(1+n)^{(V(\mathbf{y})-V(\mathbf{X}_n))}$, or set it equal to X_n otherwise. [Note that we are using $\lambda_n = \log(1+n)$.] □

12.6 The Sampling Importance Resampling Algorithm

The sampling importance resampling, or SIR, algorithm is a method for generating a random vector $\boldsymbol{X}$ whose mass function

$$f(\boldsymbol{x}) = C_1 f_o(\boldsymbol{x})$$

is specified up to a multiplicative constant by simulating a Markov chain whose limiting probabilities are given by a mass function

$$g(\boldsymbol{x}) = C_2 g_o(\boldsymbol{x})$$

that is also specified up to a multiplicative constant. It is similar to the acceptance–rejection technique, where one starts by generating the value of a random vector $\boldsymbol{Y}$ with density g and then, if $\boldsymbol{Y} = \boldsymbol{y}$, accepting this value with probability $f(\mathbf{y})/cg(\mathbf{y})$, where c is a constant chosen so that $f(\mathbf{x})/cg(\mathbf{x}) \leq 1$, for all $\boldsymbol{x}$. If the value is not accepted, then the process begins anew, and the eventually accepted value $\boldsymbol{X}$ has density f. However, as f and g are no longer totally specified, this approach is not available.

The SIR approach starts by generating m successive states of a Markov chain whose limiting probability mass function is g. Let these state values be denoted as $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$. Now, define the "weights" $w_i, i = 1, \ldots, m$, by

$$w_i = \frac{f_o(\boldsymbol{y}_i)}{g_o(\boldsymbol{y}_i)}$$

and generate a random vector $\boldsymbol{X}$ such that

$$P\{\boldsymbol{X} = \boldsymbol{y}_j\} = \frac{w_j}{\sum_{i=1}^m w_i}, \quad j = 1, \ldots, m$$

We will show that when m is large, the random vector $\boldsymbol{X}$ has a mass function approximately equal to f.

Proposition The distribution of the vector $\boldsymbol{X}$ obtained by the SIR method converges as $m \to \infty$ to f.

Proof Let $\boldsymbol{Y}_i, i = 1, \ldots, m$, denote the m random vectors generated by the Markov chain whose limiting mass function is g, and let $W_i = f_o(\boldsymbol{Y}_i)/g_o(\boldsymbol{Y}_i)$

denote their weights. For a fixed set of vectors $\mathcal{A}$, let $I_i = 1$ if $\boldsymbol{Y}_i \in \mathcal{A}$ and let it equal 0 otherwise. Then

$$P\{\boldsymbol{X} \in \mathcal{A} | \boldsymbol{Y}_i, i = 1, \dots, m\} = \frac{\sum_{i=1}^m I_i W_i}{\sum_{i=1}^m W_i} \tag{12.5}$$

Now, by the Markov chain result of Equation (12.2), we see that as $m \to \infty$,

$$\sum_{i=1}^m I_i W_i / m \to E_g[IW] = E_g[IW | I = 1] P_g\{I = 1\} = E_g[W | \boldsymbol{Y} \in \mathcal{A}] P_g\{\boldsymbol{Y} \in \mathcal{A}\}$$

and

$$\sum_{i=1}^m W_i / m \to E_g[W] = E_g[f_o(\boldsymbol{Y}) / g_o(\boldsymbol{Y})] = \int \frac{f_o(\boldsymbol{y})}{g_o(\boldsymbol{y})} g(\boldsymbol{y}) d\boldsymbol{y} = C_2 / C_1$$

Hence, dividing numerator and denominator of (12.5) by m shows that

$$P\{\boldsymbol{X} \in \mathcal{A} | \boldsymbol{Y}_i, i = 1, \dots, m\} \to \frac{C_1}{C_2} E_g[W | \boldsymbol{Y} \in \mathcal{A}] P_g\{\boldsymbol{Y} \in \mathcal{A}\}$$

But,

$$\begin{aligned}\frac{C_1}{C_2} E_g[W | \boldsymbol{Y} \in \mathcal{A}] P_g\{\boldsymbol{Y} \in \mathcal{A}\} &= \frac{C_1}{C_2} E_g\left[\frac{f_o(\boldsymbol{Y})}{g_o(\boldsymbol{Y})} | \boldsymbol{Y} \in \mathcal{A}\right] P_g\{\boldsymbol{Y} \in \mathcal{A}\} \\ &= \int_{\boldsymbol{y} \in \mathcal{A}} \frac{f(\boldsymbol{y})}{g(\boldsymbol{y})} g(\boldsymbol{y}) d\boldsymbol{y} \\ &= \int_{\boldsymbol{y} \in \mathcal{A}} f(\boldsymbol{y}) d\boldsymbol{y}\end{aligned}$$

Hence, as $m \to \infty$,

$$P\{\boldsymbol{X} \in \mathcal{A} | \boldsymbol{Y}_i, i = 1, \dots, m\} \to \int_{\boldsymbol{y} \in \mathcal{A}} f(\boldsymbol{y}) d\boldsymbol{y}$$

which implies, by a mathematical result known as Lebesgue's dominated convergence theorem, that

$$P\{\boldsymbol{X} \in \mathcal{A}\} = E[P\{\boldsymbol{X} \in \mathcal{A} | \boldsymbol{Y}_i, i = 1, \dots, m\}] \to \int_{\boldsymbol{y} \in \mathcal{A}} f(\boldsymbol{y}) d\boldsymbol{y}$$

and the result is proved. □

The sampling importance resampling algorithm for approximately generating a random vector with mass function f starts by generating random variables with a different joint mass function (as in *importance sampling*) and then *resamples* from this pool of generated values to obtain the random vector.

Suppose now that we want to estimate $E_f[h(\boldsymbol{X})]$ for some function h. This can be accomplished by first generating a large number of successive states of a Markov chain whose limiting probabilities are given by g. If these states are $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$, then it might seem natural to choose k vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$ having the probability distribution

$$P\{\boldsymbol{X} = \boldsymbol{y}_j\} = \frac{w_j}{\sum_{i=1}^m w_i}, \quad j = 1, \ldots, m$$

where k/m is small and $w_i = f_o(\boldsymbol{y}_i)/g_o(\boldsymbol{y}_i)$, and then use $\sum_{i=1}^k h(\boldsymbol{X}_i)/k$ as the estimator. However, a better approach is not to base the estimator on a sampled set of k values, but rather to use the entire set of m generated values $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$. We now show that

$$\frac{1}{\sum_{i=1}^m w_i} \sum_{j=1}^m w_j h(\boldsymbol{y}_j)$$

is a better estimator of $E_f[h(\boldsymbol{X})]$ than is $\sum_{i=1}^k h(\boldsymbol{X}_i)/k$. To show this, note that

$$E[h(\boldsymbol{X}_i)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m] = \frac{1}{\sum_{i=1}^m w_i} \sum_{j=1}^m w_j h(\boldsymbol{y}_j)$$

and thus

$$E\left[\frac{1}{k}\sum_{i=1}^k h(\boldsymbol{X}_i)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\right] = \frac{1}{\sum_{i=1}^m w_i} \sum_{j=1}^m w_j h(\boldsymbol{y}_j)$$

which shows that $\sum_{j=1}^m h(\boldsymbol{y}_j)w_j/\sum_{i=1}^m w_i$ has the same mean and smaller variance than $\sum_{i=1}^k h(\boldsymbol{X}_i)/k$.

The use of data generated from one distribution to gather information about another distribution is particularly useful in Bayesian statistics.

Example 12l Suppose that $\boldsymbol{X}$ is a random vector whose probability distribution is specified up to a vector of unknown parameters $\boldsymbol{\theta}$. For instance, $\boldsymbol{X}$ could be a sequence of independent and identically distributed normal random variables and $\boldsymbol{\theta} = (\theta_1, \theta_2)$ where θ_1 is the mean and θ_2 is the variance of these random variables. Let $f(\boldsymbol{x}|\boldsymbol{\theta})$ denote the density of $\boldsymbol{X}$ given $\boldsymbol{\theta}$. Whereas in classical statistics one assumes that $\boldsymbol{\theta}$ is a vector of unknown constants, in Bayesian statistics we suppose that it, too, is random and has a specified probability density function $p(\boldsymbol{\theta})$, called the prior density.

If $\boldsymbol{X}$ is observed to equal $\boldsymbol{x}$, then the conditional, also known as the posterior, density of $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int f(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d(\boldsymbol{\theta})}$$

However, in many situations $\int f(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d(\boldsymbol{\theta})$ cannot easily be computed, and so the preceding formula cannot be directly used to study the posterior distribution.

One approach to study the properties of the posterior distribution is to start by generating random vectors $\boldsymbol{\theta}$ from the prior density p and then use the resulting data to gather information about the posterior density $p(\boldsymbol{\theta}|x)$. If we suppose that the prior density $p(\boldsymbol{\theta})$ is completely specified and can be directly generated from, then we can use the SIR algorithm with

$$\begin{aligned} f_o(\boldsymbol{\theta}) &= f(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ g(\boldsymbol{\theta}) &= g_o(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \\ w(\boldsymbol{\theta}) &= f(\boldsymbol{x}|\boldsymbol{\theta}) \end{aligned}$$

To begin, generate a large number m of random vectors from the prior density $p(\boldsymbol{\theta})$. Let their values be $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m$. We can now estimate any function of the form $E[h(\boldsymbol{\theta})|\boldsymbol{x}]$ by the estimator

$$\sum_{j=1}^{m} \alpha_j h(\boldsymbol{\theta}_j), \quad \text{where } \alpha_j = \frac{f(\boldsymbol{x}|\boldsymbol{\theta}_j)}{\sum_{i=1}^{m} f(\boldsymbol{x}|\boldsymbol{\theta}_i)}$$

For instance, for any set $\mathcal{A}$ we would use

$$\sum_{j=1}^{m} \alpha_j I\{\boldsymbol{\theta}_j \in \mathcal{A}\} \quad \text{to estimate } P\{\boldsymbol{\theta} \in \mathcal{A}|\boldsymbol{x}\}$$

where $I\{\boldsymbol{\theta}_j \in \mathcal{A}\}$ is 1 if $\boldsymbol{\theta}_j \in \mathcal{A}$ and is 0 otherwise.

In cases where the dimension of $\boldsymbol{\theta}$ is small, we can use the generated data from the prior along with their weights to graphically explore the posterior. For instance, if $\boldsymbol{\theta}$ is two-dimensional, then we can plot the prior generated values $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m$ on a two-dimensional graph in a manner that takes the weights of these points into account. For instance, we could center a dot on each of these m points, with the area of the dot on the point $\boldsymbol{\theta}_j$ being proportional to its weight $f(\boldsymbol{x}|\boldsymbol{\theta}_j)$. Another possibility would be to let all the dots be of the same size but to let the darkness of the dot depend on its weight in a linear additive fashion. That is, for instance, if $m = 3$ and $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$, $f(\boldsymbol{x}|\boldsymbol{\theta}_3) = 2f(\boldsymbol{x}|\boldsymbol{\theta}_1)$, then the colors of the dots at $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$ should be the same.

If the prior density p is only specified up to a constant, or if it is hard to directly generate random vectors from it, then we can generate a Markov chain having p as the limiting density, and then continue as before. □

Remark Because

$$\frac{p(\theta|\mathbf{x})}{p(\theta)} = Cf(\mathbf{x}|\theta)$$

the estimator of $E[h(\theta)|\mathbf{x}]$ given in the preceding example could also have been derived by using the normalized importance sampling technique of Section 10.3.□

12.7 Coupling from the Past

Consider an irreducible Markov chain with states $1, \ldots, m$ and transition probabilities $P_{i,j}$ and suppose we want to generate the value of a random variable whose distribution is that of the stationary distribution of this Markov chain (see Section 12.1 for relevant definitions). In Section 12.1 we noted that we could *approximately* generate such a random variable by arbitrarily choosing an initial state and then simulating the resulting Markov chain for a large fixed number of time periods; the final state is used as the value of the random variable. In this section we present a procedure that generates a random variable whose distribution is **exactly** that of the stationary distribution.

If, in theory, we generated the Markov chain starting at time $-\infty$ in any arbitrary state, then the state at time 0 would have the stationary distribution. So imagine that we do this, and suppose that a different person is to generate the next state at each of these times. Thus, if $X(-n)$, the state at time $-n$, is i, then person $-n$ would generate a random variable that is equal to j with probability $P_{i,j}$, $j = 1, \ldots, m$, and the value generated would be the state at time $-(n-1)$. Now suppose that person -1 wants to do his random variable generation early. Because he does not know what the state at time -1 will be, he generates a sequence of random variables $N_{-1}(i), i = 1, \ldots, m$, where $N_{-1}(i)$, the next state if $X(-1) = i$, is equal to j with probability $P_{i,j}$, $j = 1, \ldots, m$. If it results that $X(-1) = i$, then person -1 would report that the state at time 0 is

$$S_{-1}(i) = N_{-1}(i), \quad i = 1, \ldots, m$$

(That is, $S_{-1}(i)$ is the simulated state at time 0 when the simulated state at time -1 is i.)

Now suppose that person -2, hearing that person -1 is doing his simulation early, decides to do the same thing. She generates a sequence of random variables $N_{-2}(i), i = 1, \ldots, m$, where $N_{-2}(i)$ is equal to j with probability $P_{i,j}$, $j = 1, \ldots, m$. Consequently, if it is reported to her that $X(-2) = i$, then she will report that $X(-1) = N_{-2}(i)$. Combining this with the early generation of person -1 shows that if $X(-2) = i$, then the simulated state at time 0 is

$$S_{-2}(i) = S_{-1}(N_{-2}(i)), \quad i = 1, \ldots, m$$

Continuing in the preceding manner, suppose that person -3 generates a sequence of random variables $N_{-3}(i), i = 1, \ldots, m$, where $N_{-3}(i)$ is to be the generated value of the next state when $X(-3) = i$. Consequently, if $X(-3) = i$ then the simulated state at time 0 would be

$$S_{-3}(i) = S_{-2}(N_{-3}(i)), \quad i = 1, \ldots, m$$

Now suppose we continue the preceding, and so obtain the simulated functions

$$S_{-1}(i), S_{-2}(i), S_{-3}(i), \ldots \quad i = 1, \ldots, m$$

Going backwards in time in this manner, we will at sometime, say $-r$, have a simulated function $S_{-r}(i)$ that is a constant function. That is, for some state j, $S_{-r}(i)$ will equal j for all states $i = 1, \ldots, m$. But this means that no matter what the simulated values from time $-\infty$ to $-r$, we can be certain that the simulated value at time 0 is j. Consequently, j can be taken as the value of a generated random variable whose distribution is exactly that of the stationary distribution of the Markov chain.

Example 12m Consider a Markov chain with states 1, 2, 3 and suppose that simulation yielded the values

$$N_{-1}(i) = \begin{cases} 3, & \text{if } i = 1 \\ 2, & \text{if } i = 2 \\ 2, & \text{if } i = 3 \end{cases}$$

and

$$N_{-2}(i) = \begin{cases} 1, & \text{if } i = 1 \\ 3, & \text{if } i = 2 \\ 1, & \text{if } i = 3 \end{cases}$$

Then

$$S_{-2}(i) = \begin{cases} 3, & \text{if } i = 1 \\ 2, & \text{if } i = 2 \\ 3, & \text{if } i = 3 \end{cases}$$

If

$$N_{-3}(i) = \begin{cases} 3, & \text{if } i = 1 \\ 1, & \text{if } i = 2 \\ 1, & \text{if } i = 3 \end{cases}$$

then

$$S_{-3}(i) = \begin{cases} 3, & \text{if } i = 1 \\ 3, & \text{if } i = 2 \\ 3, & \text{if } i = 3 \end{cases}$$

Therefore, no matter what the state is at time -3, the state at time 0 will be 3. □

Remarks The procedure developed in this section for generating a random variable whose distribution is the stationary distribution of the Markov chain is called *coupling from the past*. □

Exercises

1. Let π_j, $j = 1, \ldots, N$, denote the stationary probabilities of a Markov chain. Show that if $P\{X_0 = j\} = \pi_j$, $j = 1, \ldots, N$, then

$$P\{X_n = j\} = \pi_j, \quad \text{for all } n, j$$

2. Let $\mathbf{Q}$ be a symmetric transition probability matrix, that is, $q_{ij} = q_{ji}$ for all i, j. Consider a Markov chain which, when the present state is i, generates the value of a random variable X such that $P\{X = j\} = q_{ij}$, and if $X = j$, then either moves to state j with probability $b_j/(b_i + b_j)$, or remains in state i otherwise, where $b_j, j = 1 \ldots, N$, are specified positive numbers. Show that the resulting Markov chain is time reversible with limiting probabilities $\pi_j = Cb_j, j = 1, \ldots, N$.

3. Let $\pi_i, i = 1, \ldots, n$ be positive numbers that sum to 1. Let $\mathbf{Q}$ be an irreducible transition probability matrix with transition probabilities $q(i, j), i, j = 1, \ldots, n$. Suppose that we simulate a Markov chain in the following manner: if the current state of this chain is i, then we generate a random variable that is equal to k with probability $q(i, k), k = 1, \ldots, n$. If the generated value is j then the next state of the Markov chain is either i or j, being equal to j with probability $\frac{\pi_j q(j,i)}{\pi_i q(i,j)+\pi_j q(j,i)}$ and to i with probability $1 - \frac{\pi_j q(j,i)}{\pi_i q(i,j)+\pi_j q(j,i)}$.

   (a) Give the transition probabilities of the Markov chain we are simulating.
   (b) Show that $\{\pi_1, \ldots, \pi_n\}$ are the stationary probabilities of this chain.

4. Explain how to use a Markov chain monte carlo method to generate the value of a random vector $X_1, \ldots, X_{10}$ whose distribution is approximately the conditional distribution of 10 independent exponential random variables with common mean 1 given that $\prod_{i=1}^{10} X_i > 20$.

5. Let $U_1, \ldots, U_n$ be independent uniform (0, 1) random variables. For constants $a_1 > a_2 > \ldots > a_n > 0$ give a method for generating a random vector whose distribution is approximately that of the conditional distribution of $U_1, \ldots, U_n$ given that $a_1 U_1 < a_2 U_2 < \ldots < a_n U_n$.

6. Suppose that the random variables X and Y both take on values in the interval $(0, B)$. Suppose that the joint density of X given that $Y = y$ is

$$f(x|y) = C(y)e^{-xy}, \quad 0 < x < B$$

   and the joint density of Y given that $X = x$ is

$$f(y|x) = C(x)e^{-xy}, \quad 0 < y < B$$

   Give a method for approximately simulating the vector X, Y. Run a simulation to estimate (a) $E[X]$ and (b) $E[XY]$.

7. Give an efficient method for generating nine uniform points on (0, 1) conditional on the event than no two of them are within 0.1 of each other. (It can be shown that if n points are independent and uniformly distributed on

(0, 1), then the probability that no two of them are within d of each other is, for $0 < d < 1/(n-1)$, $[1-(n-1)d]^n$.)

8. In Example 12d, it can be shown that the limiting mass function of the number of customers at the $m+1$ servers is

$$p(n_1, \ldots, n_m, n_{m+1}) = C\prod_{i=1}^{m+1} P_i(n_i), \quad \sum_{i=1}^{m+1} n_i = r$$

where for each $i = 1, \ldots, m+1$, $P_i(n)$, $n = 0, \ldots, r$, is a probability mass function. Let $\mathbf{e}_k$ be the $m+1$ component vector with a 1 in the kth position and zeros elsewhere. For a vector $\mathbf{n} = (n_1, \ldots, n_{m+1})$, let

$$q(\mathbf{n}, \mathbf{n}-\mathbf{e_i}+\mathbf{e_j}) = \frac{I(n_i > 0)}{(m+1)\sum_{j=1}^{m+1} I(n_j > 0)}$$

In words, q is the transition probability matrix of a Markov chain that at each step randomly selects a nonempty server and then sends one of its customers to a randomly chosen server. Using this q function, give the Hastings–Metropolis algorithm for generating a Markov chain having $p(n_1, \ldots, n_m, n_{m+1})$ as its limiting mass function.

9. Let X_i, $i = 1, 2, 3$, be independent exponentials with mean 1. Run a simulation study to estimate

(a) $E[X_1 + 2X_2 + 3X_3 | X_1 + 2X_2 + 3X_3 > 15]$.
(b) $E[X_1 + 2X_2 + 3X_3 | X_1 + 2X_2 + 3X_3 < 1]$.

10. A random selection of m balls is to be made from an urn that contains n balls, n_i of which have color type $i = 1, \ldots, r$, $\sum_{i=1}^r n_i = n$. Let X_i denote the number of withdrawn balls that have color type i. Give an efficient procedure for simulating $X_1, \ldots, X_r$ conditional on the event that all r color types are represented in the random selection. Assume that the probability that all color types are represented in the selection is a small positive number.

11. Suppose the joint density of X, Y, Z is given by

$$f(x, y, z) = Ce^{-(x+y+z+axy+bxz+cyz)}, \quad x > 0,\ y > 0,\ z > 0$$

where a, b, c are specified nonnegative constants, and C does not depend on x, y, z. Explain how we can simulate the vector X, Y, Z, and run a simulation to estimate $E[XYZ]$ when $a = b = c = 1$.

12. Suppose that for random variables X, Y, N

$$P\{X = i, y \leq Y \leq y + dy, N = n\}$$
$$\approx C\binom{n}{i} y^{i+\alpha-1}(1-y)^{ni+\beta-1}e^{-\lambda}\frac{\lambda^n}{n!}dy$$

where $i = 0, \ldots, n, n = 0, 1, \ldots, y \geq 0$, and where α, β, λ are specified constants. Run a simulation to estimate $E[X]$, $E[Y]$, and $E[N]$ when $\alpha = 2, \beta = 3, \lambda = 4$.

13. Use the SIR algorithm to generate a permutation of $1, 2, \ldots, 100$ whose distribution is approximately that of a random permutation $X_1, \ldots, X_{100}$ conditioned on the event that $\sum_j jX_j > 285, 000$.

14. Let $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n$ be random points in ℓ, the circle of radius 1 centered at the origin. Suppose that for some r, $0 < r < 1$, their joint density function is given by

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = K \ \exp\{-\beta t(r : \mathbf{x}_1, \ldots, \mathbf{x}_n)\}, \quad \mathbf{x}_i \in \ell, \ i = 1, \ldots, n$$

where $t(r : \mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the number of the $\binom{n}{2}$ pairs of points $\mathbf{x}_i, \mathbf{x}_j, i \neq j$, that are within a distance r of each other, and $0 < \beta < \infty$. (Note that $\beta = \infty$ corresponds to the case where the $\mathbf{X}^i$ are uniformly distributed on the circle subject to the constraint that no two points are within a distance r of each other.) Explain how you can use the SIR algorithm to approximately generate these random points. If r and β were both large, would this be an efficient algorithm?

15. Generate 100 random numbers $U_{0,k}, k = 1, \ldots, 10, U_{i,j}, i \neq j, i, j = 1, \ldots, 10$. Now, consider a traveling salesman problem in which the salesman starts at city 0 and must travel in turn to each of the 10 cities $1, \ldots, 10$ according to some permutation of $1, \ldots, 10$. Let U_{ij} be the reward earned by the salesman when he goes directly from city i to city j. Use simulated annealing to approximate the maximal possible return of the salesman.

Bibliography

Aarts, E., and J. Korst, *Simulated Annealing and Boltzmann Machines*. Wiley, New York, 1989.

Besag, J., "Towards Bayesian Image Analysis," *J. Appl. Statistics*, **16**, 395–407, 1989.

Besag, J., P. Green, D. Higdon, and K. Mengersen, "Bayesian Computation and Stochastic Systems (with Discussion)," *Statistical Sci.*, **10**, 3–67, 1995.

Diaconis, P., and S. Holmes, "Three Examples of Monte-Carlo Markov Chains: At the Interface between Statistical Computing, Computer Science, and Statistical Mechanics," *Discrete Probability and Algorithms* (D. Aldous, P. Diaconis, J. Spence, and J. M. Steele, eds.), pp. 43–56. Springer-Verlag, 1995.

Gelfand, A. E., S. E. Hills, A. Racine-Poon, and A. F. Smith, "Illustration of Bayesian Inference in Normal Data Models using Gibbs Sampling," *J. Am. Statistical Assoc.*, **85**, 972–985, 1990.

Gelfand, A. E., and A. F. Smith, "Sampling Based Approaches to Calculating Marginal Densities," *J. Am. Statistical Assoc.*, **85**, 398–409, 1990.

Gelman, A., and D. B. Rubin, "Inference from Iterative Simulation (with Discussion)," *Statistical Sci.*, **7**, 457–511, 1992.

Geman, S., and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Machine Intelligence*, **6**, 721–724, 1984.

Geyer, C. J., "Practical Markov Chain Monte Carlo (with Discussion)," *Statistical Sci.*, **7**, 473–511, 1992.

Gidas, B., "Metropolis-type Monte Carlo Simulation Algorithms and Simulated Annealing," in *Trends in Contemporary Probability* (J. L. Snell, ed.). CRC Press. Boca Raton, FL, 1995.

Hajek, B., "Cooling Schedules for Optimal Annealing," *Math. Operations Res.*, **13**, **311–329**, 1989.

Hammersley, J. M., and D. C. Handscomb, *Monte Carlo Methods.* Methuen, London, 1965.

Ripley, B., *Stochastic Simulation.* Wiley, New York, 1987.

Rubin, D. R., "Using the SIR Algorithm to Simulate Posterior Distributions," in *Bayesian Statistics 3* (J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, eds.), pp. 395–402. Oxford University Press, 1988.

Rubinstein, R. R., *Monte Carlo Optimization, Simulation, and Sensitivity of Queueing Networks.* Wiley, New York, 1986.

Sinclair, A., *Algorithms for Random Generation and Counting.* Birkhauser, Boston, 1993.

Smith, A. F., and G. O. Roberts, "Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods (with Discussion)," *J. Roy. Statistical Soc., Ser. B*, **55**, 3–23, 1993.