Simulation

11.1 Introduction

Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote a random vector having a given density function $f(x_1, \ldots, x_n)$ and suppose we are interested in computing

$$E[g(\mathbf{X})] = \iint \cdots \int g(x_1, \ldots, x_n) f(x_1, \ldots, x_n)\, dx_1\, dx_2 \cdots dx_n$$

for some n-dimensional function g. For instance, g could represent the total delay in queue of the first $[n/2]$ customers when the X values represent the first $[n/2]$ interarrival and service times.* In many situations, it is not analytically possible either to compute the preceding multiple integral exactly or even to numerically approximate it within a given accuracy. One possibility that remains is to approximate $E[g(\mathbf{X})]$ by means of simulation.

To approximate $E[g(\mathbf{X})]$, start by generating a random vector $\mathbf{X}^{(1)} = \left(X_1^{(1)}, \ldots, X_n^{(1)}\right)$ having the joint density $f(x_1, \ldots, x_n)$ and then compute $Y^{(1)} = g\left(\mathbf{X}^{(1)}\right)$. Now generate a second random vector (independent of the first) $\mathbf{X}^{(2)}$ and compute $Y^{(2)} = g\left(\mathbf{X}^{(2)}\right)$. Keep on doing this until r, a fixed number of independent and identically distributed random variables $Y^{(i)} = g\left(\mathbf{X}^{(i)}\right)$,

*We are using the notation $[a]$ to represent the largest integer less than or equal to a.

Introduction to Probability Models, ISBN: 9780123756862

$i = 1, \ldots, r$ have been generated. Now by the strong law of large numbers, we know that

$$\lim_{r\to\infty} \frac{Y^{(1)} + \cdots + Y^{(r)}}{r} = E\left[Y^{(i)}\right] = E[g(\mathbf{X})]$$

and so we can use the average of the generated Ys as an estimate of $E[g(\mathbf{X})]$. This approach to estimating $E[g(\mathbf{X})]$ is called the *Monte Carlo simulation* approach.

Clearly there remains the problem of how to generate, or *simulate*, random vectors having a specified joint distribution. The first step in doing this is to be able to generate random variables from a uniform distribution on $(0, 1)$. One way to do this would be to take 10 identical slips of paper, numbered $0, 1, \ldots, 9$, place them in a hat and then successively select n slips, with replacement, from the hat. The sequence of digits obtained (with a decimal point in front) can be regarded as the value of a uniform $(0, 1)$ random variable rounded off to the nearest $\left(\frac{1}{10}\right)^n$. For instance, if the sequence of digits selected is 3, 8, 7, 2, 1, then the value of the uniform $(0, 1)$ random variable is 0.38721 (to the nearest 0.00001). Tables of the values of uniform $(0, 1)$ random variables, known as random number tables, have been extensively published (for instance, see The RAND Corporation, *A Million Random Digits with 100,000 Normal Deviates* (New York: The Free Press, 1955)). Table 11.1 is such a table.

However, this is not the way in which digital computers simulate uniform $(0, 1)$ random variables. In practice, they use pseudo random numbers instead of truly random ones. Most random number generators start with an initial value X_0, called the seed, and then recursively compute values by specifying positive integers a, c, and m, and then letting

$$X_{n+1} = (aX_n + c) \quad \text{modulo } m, \quad n \geqslant 0$$

where the preceding means that $aX_n + c$ is divided by m and the remainder is taken as the value of X_{n+1}. Thus each X_n is either $0, 1, \ldots,$ or $m - 1$ and the quantity X_n/m is taken as an approximation to a uniform $(0, 1)$ random variable. It can be shown that subject to suitable choices for a, c, m, the preceding gives rise to a sequence of numbers that looks as if it were generated from independent uniform $(0, 1)$ random variables.

As our starting point in the simulation of random variables from an arbitrary distribution, we shall suppose that we can simulate from the uniform $(0, 1)$ distribution, and we shall use the term "random numbers" to mean independent random variables from this distribution. In Sections 11.2 and 11.3 we present both general and special techniques for simulating continuous random variables; and in Section 11.4 we do the same for discrete random variables. In Section 11.5 we discuss the simulation both of jointly distributed random variables and stochastic processes. Particular attention is given to the simulation of nonhomogeneous

Table 11.1 A Random Number Table

04839	96423	24878	82651	66566	14778	76797	14780	13300	87074
68086	26432	46901	20848	89768	81536	86645	12659	92259	57102
39064	66432	84673	40027	32832	61362	98947	96067	64760	64584
25669	26422	44407	44048	37937	63904	45766	66134	75470	66520
64117	94305	26766	25940	39972	22209	71500	64568	91402	42416
87917	77341	42206	35126	74087	99547	81817	42607	43808	76655
62797	56170	86324	88072	76222	36086	84637	93161	76038	65855
95876	55293	18988	27354	26575	08625	40801	59920	29841	80150
29888	88604	67917	48708	18912	82271	65424	69774	33611	54262
73577	12908	30883	18317	28290	35797	05998	41688	34952	37888
27958	30134	04024	86385	29880	99730	55536	84855	29080	09250
90999	49127	20044	59931	06115	20542	18059	02008	73708	83517
18845	49618	02304	51038	20655	58727	28168	15475	56942	53389
94824	78171	84610	82834	09922	25417	44137	48413	25555	21246
35605	81263	39667	47358	56873	56307	61607	49518	89356	20103
33362	64270	01638	92477	66969	98420	04880	45585	46565	04102
88720	82765	34476	17032	87589	40836	32427	70002	70663	88863
39475	46473	23219	53416	94970	25832	69975	94884	19661	72828
06990	67245	68350	82948	11398	42878	80287	88267	47363	46634
40980	07391	58745	25774	22987	80059	39911	96189	41151	14222
83974	29992	65381	38857	50490	83765	55657	14361	31720	57375
33339	31926	14883	24413	59744	92351	97473	89286	35931	04110
31662	25388	61642	34072	81249	35648	56891	69352	48373	45578
93526	70765	10592	04542	76463	54328	02349	17247	28865	14777
20492	38391	91132	21999	59516	81652	27195	48223	46751	22923
04153	53381	79401	21438	83035	92350	36693	31238	59649	91754
05520	91962	04739	13092	97662	24822	94730	06496	35090	04822
47498	87637	99016	71060	88824	71013	18735	20286	23153	72924
23167	49323	45021	33132	12544	41035	80780	45393	44812	12515
23792	14422	15059	45799	22716	19792	09983	74353	68668	30429
85900	98275	32388	52390	16815	69298	82732	38480	73817	32523
42559	78985	05300	22164	24369	54224	35083	19687	11062	91491
14349	82674	66523	44133	00697	35552	35970	19124	63318	29686
17403	53363	44167	64486	64758	75366	76554	31601	12614	33072
23632	27889	47914	02584	37680	20801	72152	39339	34806	08930

Poisson processes, and in fact three different approaches for this are discussed. Simulation of two-dimensional Poisson processes is discussed in Section 11.5.2. In Section 11.6 we discuss various methods for increasing the precision of the simulation estimates by reducing their variance; and in Section 11.7 we consider the problem of choosing the number of simulation runs needed to attain a desired level of precision. Before beginning this program, however, let us consider two applications of simulation to combinatorial problems.

Example 11.1 (Generating a Random Permutation) Suppose we are interested in generating a permutation of the numbers $1, 2, \ldots, n$ that is such that all $n!$ possible orderings are equally likely. The following algorithm will accomplish this by first choosing one of the numbers $1, \ldots, n$ at random and then putting that number in position n; it then chooses at random one of the remaining $n - 1$ numbers and puts that number in position $n - 1$; it then chooses at random one of the remaining $n - 2$ numbers and puts it in position $n - 2$, and so on (where choosing a number at random means that each of the remaining numbers is equally likely to be chosen). However, so that we do not have to consider exactly which of the numbers remain to be positioned, it is convenient and efficient to keep the numbers in an ordered list and then randomly choose the position of the number rather than the number itself. That is, starting with any initial ordering $p_1, p_2, \ldots, p_n$, we pick one of the positions $1, \ldots, n$ at random and then interchange the number in that position with the one in position n. Now we randomly choose one of the positions $1, \ldots, n - 1$ and interchange the number in this position with the one in position $n - 1$, and so on.

To implement the preceding, we need to be able to generate a random variable that is equally likely to take on any of the values $1, 2, \ldots, k$. To accomplish this, let U denote a random number—that is, U is uniformly distributed over $(0, 1)$—and note that kU is uniform on $(0, k)$ and so

$$P\{i - 1 < kU < i\} = \frac{1}{k}, \quad i = 1, \ldots, k$$

Hence, the random variable $I = [kU] + 1$ will be such that

$$P\{I = i\} = P\{[kU] = i - 1\} = P\{i - 1 < kU < i\} = \frac{1}{k}$$

The preceding algorithm for generating a random permutation can now be written as follows:

Step 1: Let $p_1, p_2, \ldots, p_n$ be any permutation of $1, 2, \ldots, n$ (for instance, we can choose $p_j = j, j = 1, \ldots, n$).
Step 2: Set $k = n$.
Step 3: Generate a random number U and let $I = [kU] + 1$.
Step 4: Interchange the values of p_I and p_k.
Step 5: Let $k = k - 1$ and if $k > 1$ go to step 3.
Step 6: $p_1, \ldots, p_n$ is the desired random permutation.

For instance, suppose $n = 4$ and the initial permutation is 1, 2, 3, 4. If the first value of I (which is equally likely to be either 1, 2, 3, 4) is $I = 3$, then the new permutation is 1, 2, 4, 3. If the next value of I is $I = 2$ then the new permutation is 1, 4, 2, 3. If the final value of I is $I = 2$, then the final permutation is 1, 4, 2, 3, and this is the value of the random permutation.

One very important property of the preceding algorithm is that it can also be used to generate a random subset, say of size r, of the integers $1, \ldots, n$. Namely, just follow the algorithm until the positions $n, n-1, \ldots, n-r+1$ are filled. The elements in these positions constitute the random subset. ■

Example 11.2 (Estimating the Number of Distinct Entries in a Large List) Consider a list of n entries where n is very large, and suppose we are interested in estimating d, the number of distinct elements in the list. If we let m_i denote the number of times that the element in position i appears on the list, then we can express d by

$$d = \sum_{i=1}^{n} \frac{1}{m_i}$$

To estimate d, suppose that we generate a random value X equally likely to be either $1, 2, \ldots, n$ (that is, we take $X = [nU] + 1$) and then let $m(X)$ denote the number of times the element in position X appears on the list. Then

$$E\left[\frac{1}{m(X)}\right] = \sum_{i=1}^{n} \frac{1}{m_i}\frac{1}{n} = \frac{d}{n}$$

Hence, if we generate k such random variables $X_1, \ldots, X_k$ we can estimate d by

$$d \approx \frac{n\sum_{i=1}^{k} 1/m(X_i)}{k}$$

Suppose now that each item in the list has a value attached to it—$v(i)$ being the value of the ith element. The sum of the values of the distinct items—call it v—can be expressed as

$$v = \sum_{i=1}^{n} \frac{v(i)}{m(i)}$$

Now if $X = [nU] + 1$, where U is a random number, then

$$E\left[\frac{v(X)}{m(X)}\right] = \sum_{i=1}^{n} \frac{v(i)}{m(i)}\frac{1}{n} = \frac{v}{n}$$

Hence, we can estimate v by generating $X_1, \ldots, X_k$ and then estimating v by

$$v \approx \frac{n}{k}\sum_{i=1}^{k} \frac{v(X_i)}{m(X_i)}$$

For an important application of the preceding, let $A_i = \{a_{i,1}, \ldots, a_{i,n_i}\}, i = 1, \ldots, s$ denote events, and suppose we are interested in estimating $P\left(\bigcup_{i=1}^{s} A_i\right)$. Since

$$P\left(\bigcup_{i=1}^{s} A_i\right) = \sum_{a \in \cup A_i} P(a) = \sum_{i=1}^{s} \sum_{j=1}^{n_i} \frac{P(a_{i,j})}{m(a_{i,j})}$$

where $m(a_{i,j})$ is the number of events to which the point $a_{i,j}$ belongs, the preceding method can be used to estimate $P\left(\bigcup_1^s A_i\right)$.

Note that the preceding procedure for estimating v can be effected without prior knowledge of the set of values $\{v_1, \ldots, v_n\}$. That is, it suffices that we can determine the value of an element in a specific place and the number of times that element appears on the list. When the set of values is *a priori* known, there is another approach available as will be shown in Example 11.11. ■

11.2 General Techniques for Simulating Continuous Random Variables

In this section we present three methods for simulating continuous random variables.

11.2.1 The Inverse Transformation Method

A general method for simulating a random variable having a continuous distribution—called the *inverse transformation method*—is based on the following proposition.

Proposition 11.1 Let U be a uniform $(0, 1)$ random variable. For any continuous distribution function F if we define the random variable X by

$$X = F^{-1}(U)$$

then the random variable X has distribution function F. ($F^{-1}(u)$ is defined to equal that value x for which $F(x) = u$.)

Proof.

$$\begin{aligned} F_X(a) &= P\{X \leqslant a\} \\ &= P\{F^{-1}(U) \leqslant a\} \end{aligned} \tag{11.1}$$

Now, since $F(x)$ is a monotone function, it follows that $F^{-1}(U) \leqslant a$ if and only if $U \leqslant F(a)$. Hence, from Equation (11.1), we see that

$$\begin{aligned} F_X(a) &= P\{U \leqslant F(a)\} \\ &= F(a) \end{aligned}$$

■

Hence, we can simulate a random variable X from the continuous distribution F, when F^{-1} is computable, by simulating a random number U and then setting $X = F^{-1}(U)$.

Example 11.3 (Simulating an Exponential Random Variable) If $F(x) = 1 - e^{-x}$, then $F^{-1}(u)$ is that value of x such that

$$1 - e^{-x} = u$$

or

$$x = -\log(1 - u)$$

Hence, if U is a uniform $(0, 1)$ variable, then

$$F^{-1}(U) = -\log(1 - U)$$

is exponentially distributed with mean 1. Since $1 - U$ is also uniformly distributed on $(0, 1)$ it follows that $-\log U$ is exponential with mean 1. Since cX is exponential with mean c when X is exponential with mean 1, it follows that $-c \log U$ is exponential with mean c. ■

11.2.2 The Rejection Method

Suppose that we have a method for simulating a random variable having density function $g(x)$. We can use this as the basis for simulating from the continuous distribution having density $f(x)$ by simulating Y from g and then accepting this simulated value with a probability proportional to $f(Y)/g(Y)$.

Specifically, let c be a constant such that

$$\frac{f(y)}{g(y)} \leqslant c \quad \text{for all } y$$

We then have the following technique for simulating a random variable having density f.

Rejection Method

Step 1: Simulate Y having density g and simulate a random number U.
Step 2: If $U \leqslant f(Y)/cg(Y)$ set $X = Y$. Otherwise return to step 1.

Proposition 11.2 The random variable X generated by the rejection method has density function f.

Proof. Let X be the value obtained, and let N denote the number of necessary iterations. Then

$$\begin{aligned} P\{X \leqslant x\} &= P\{Y_N \leqslant x\} \\ &= P\{Y \leqslant x | U \leqslant f(Y)/cg(Y)\} \\ &= \frac{P\{Y \leqslant x, U \leqslant f(Y)/cg(Y)\}}{K} \\ &= \frac{\int P\{Y \leqslant x, U \leqslant f(Y)/cg(Y) | Y = y\} g(y)\, dy}{K} \\ &= \frac{\int_{-\infty}^{x} (f(y)/cg(y)) g(y)\, dy}{K} \\ &= \frac{\int_{-\infty}^{x} f(y)\, dy}{Kc} \end{aligned}$$

where $K = P\{U \leqslant f(Y)/cg(Y)\}$. Letting $x \to \infty$ shows that $K = 1/c$ and the proof is complete. ■

Remarks

(i) The preceding method was originally presented by Von Neumann in the special case where g was positive only in some finite interval (a, b), and Y was chosen to be uniform over (a, b) (that is, $Y = a + (b - a)U$).

(ii) Note that the way in which we "accept the value Y with probability $f(Y)/cg(Y)$" is by generating a uniform $(0, 1)$ random variable U and then accepting Y if $U \leqslant f(Y)/cg(Y)$.

(iii) Since each iteration of the method will, independently, result in an accepted value with probability $P\{U \leqslant f(Y)/cg(Y)\} = 1/c$ it follows that the number of iterations is geometric with mean c.

(iv) Actually, it is not necessary to generate a new uniform random number when deciding whether or not to accept, since at a cost of some additional computation, a single random number, suitably modified at each iteration, can be used throughout. To see how, note that the actual value of U is not used—only whether or not $U < f(Y)/cg(Y)$. Hence, if Y is rejected—that is, if $U > f(Y)/cg(Y)$—we can use the fact that, given Y,

$$\frac{U - f(Y)/cg(Y)}{1 - f(Y)/cg(Y)} = \frac{cUg(Y) - f(Y)}{cg(Y) - f(Y)}$$

is uniform on $(0, 1)$. Hence, this may be used as a uniform random number in the next iteration. As this saves the generation of a random number at the cost of the preceding computation, whether it is a net savings depends greatly upon the method being used to generate random numbers.

Example 11.4 Let us use the rejection method to generate a random variable having density function

$$f(x) = 20x(1-x)^3, \quad 0 < x < 1$$

Since this random variable (which is beta with parameters 2, 4) is concentrated in the interval $(0, 1)$, let us consider the rejection method with

$$g(x) = 1, \quad 0 < x < 1$$

To determine the constant c such that $f(x)/g(x) \leqslant c$, we use calculus to determine the maximum value of

$$\frac{f(x)}{g(x)} = 20x(1-x)^3$$

Differentiation of this quantity yields

$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = 20\left[(1-x)^3 - 3x(1-x)^2\right]$$

Setting this equal to 0 shows that the maximal value is attained when $x = \frac{1}{4}$, and thus

$$\frac{f(x)}{g(x)} \leqslant 20\left(\frac{1}{4}\right)\left(\frac{3}{4}\right)^3 = \frac{135}{64} \equiv c$$

Hence,

$$\frac{f(x)}{cg(x)} = \frac{256}{27}\, x(1-x)^3$$

and thus the rejection procedure is as follows:

Step 1: Generate random numbers U_1 and U_2.
Step 2: If $U_2 \leqslant \frac{256}{27} U_1(1-U_1)^3$, stop and set $X = U_1$. Otherwise return to step 1.

The average number of times that step 1 will be performed is $c = \frac{135}{64}$. ■

Example 11.5 (Simulating a Normal Random Variable) To simulate a standard normal random variable Z (that is, one with mean 0 and variance 1) note first that the absolute value of Z has density function

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad 0 < x < \infty \tag{11.2}$$

We will start by simulating from the preceding density by using the rejection method with

$$g(x) = e^{-x}, \quad 0 < x < \infty$$

Now, note that

$$\frac{f(x)}{g(x)} = \sqrt{2e/\pi}\exp\{-(x-1)^2/2\} \leqslant \sqrt{2e/\pi}$$

Hence, using the rejection method we can simulate from Equation (11.2) as follows:

(a) Generate independent random variables Y and U, Y being exponential with rate 1 and U being uniform on $(0, 1)$.
(b) If $U \leqslant \exp\{-(Y-1)^2/2\}$, or equivalently, if

$$-\log\ U \geqslant (Y-1)^2/2$$

set $X = Y$. Otherwise return to step (a).

Once we have simulated a random variable X having Density Function (11.2) we can then generate a standard normal random variable Z by letting Z be equally likely to be either X or $-X$.

To improve upon the foregoing, note first that from Example 11.3 it follows that $-\log\ U$ will also be exponential with rate 1. Hence, steps (a) and (b) are equivalent to the following:

(a′) Generate independent exponentials with rate 1, Y_1, and Y_2.
(b′) Set $X = Y_1$ if $Y_2 \geqslant (Y_1 - 1)^2/2$. Otherwise return to step (a′).

Now suppose that we accept step (b′). It then follows by the lack of memory property of the exponential that the amount by which Y_2 exceeds $(Y_1 - 1)^2/2$ will also be exponential with rate 1.

Hence, summing up, we have the following algorithm which generates an exponential with rate 1 and an independent standard normal random variable:

Step 1: Generate Y_1, an exponential random variable with rate 1.
Step 2: Generate Y_2, an exponential with rate 1.
Step 3: If $Y_2 - (Y_1 - 1)^2/2 > 0$, set $Y = Y_2 - (Y_1 - 1)^2/2$ and go to step 4. Otherwise go to step 1.
Step 4: Generate a random number U and set

$$Z = \begin{cases} Y_1, & \text{if } U \leqslant \frac{1}{2} \\ -Y_1, & \text{if } U > \frac{1}{2} \end{cases}$$

The random variables Z and Y generated by the preceding are independent with Z being normal with mean 0 and variance 1 and Y being exponential with rate 1.

(If we want the normal random variable to have mean μ and variance σ^2, just take $\mu + \sigma Z$.) ■

Remarks

(i) Since $c = \sqrt{2e/\pi} \approx 1.32$, the preceding requires a geometric distributed number of iterations of step 2 with mean 1.32.

(ii) The final random number of step 4 need not be separately simulated but rather can be obtained from the first digit of any random number used earlier. That is, suppose we generate a random number to simulate an exponential; then we can strip off the initial digit of this random number and just use the remaining digits (with the decimal point moved one step to the right) as the random number. If this initial digit is 0, 1, 2, 3, or 4 (or 0 if the computer is generating binary digits), then we take the sign of Z to be positive and take it to be negative otherwise.

(iii) If we are generating a sequence of standard normal random variables, then we can use the exponential obtained in step 4 as the initial exponential needed in step 1 for the next normal to be generated. Hence, on the average, we can simulate a unit normal by generating 1.64 exponentials and computing 1.32 squares.

11.2.3 The Hazard Rate Method

Let F be a continuous distribution function with $\bar{F}(0) = 1$. Recall that $\lambda(t)$, the hazard rate function of F, is defined by

$$\lambda(t) = \frac{f(t)}{\bar{F}(t)}, \quad t \geqslant 0$$

(where $f(t) = F'(t)$ is the density function). Recall also that $\lambda(t)$ represents the instantaneous probability intensity that an item having life distribution F will fail at time t given it has survived to that time.

Suppose now that we are given a bounded function $\lambda(t)$, such that $\int_0^\infty \lambda(t)\, dt = \infty$, and we desire to simulate a random variable S having $\lambda(t)$ as its hazard rate function.

To do so let λ be such that

$$\lambda(t) \leqslant \lambda \quad \text{for all } t \geqslant 0$$

To simulate from $\lambda(t), t \geqslant 0$, we will

(a) simulate a Poisson process having rate λ. We will then only "accept" or "count" certain of these Poisson events. Specifically we will

(b) count an event that occurs at time t, independently of all else, with probability $\lambda(t)/\lambda$.

We now have the following proposition.

Proposition 11.3 The time of the first counted event—call it S—is a random variable whose distribution has hazard rate function $\lambda(t), t \geqslant 0$.

Proof.

$$
\begin{aligned}
&P\{t < S < t + dt | S > t\} \\
&\quad = P\{\text{first counted event in } (t, t + dt) | \text{no counted events prior to } t\} \\
&\quad = P\{\text{Poisson event in } (t, t + dt), \text{it is counted} | \text{no counted events prior to } t\} \\
&\quad = P\{\text{Poisson event in } (t, t + dt), \text{it is counted}\} \\
&\quad = [\lambda\, dt + o(dt)]\frac{\lambda(t)}{\lambda} = \lambda(t)\, dt + o(dt)
\end{aligned}
$$

which completes the proof. Note that the next to last equality follows from the independent increment property of Poisson processes. ■

Because the interarrival times of a Poisson process having rate λ are exponential with rate λ, it thus follows from Example 11.3 and the previous proposition that the following algorithm will generate a random variable having hazard rate function $\lambda(t), t \geqslant 0$.

Hazard Rate Method for Generating S: $\lambda_S(t) = \lambda(t)$

Let λ be such that $\lambda(t) \leqslant \lambda$ for all $t \geqslant 0$. Generate pairs of random variables $U_i, X_i, i \geqslant 1$, with X_i being exponential with rate λ and U_i being uniform $(0, 1)$, stopping at

$$N = \min\left\{n: U_n \leqslant \lambda\left(\sum_{i=1}^{n} X_i\right)\Big/\lambda\right\}$$

Set

$$S = \sum_{i=1}^{N} X_i$$ ■

To compute $E[N]$ we need the result, known as Wald's equation, which states that if $X_1, X_2, \ldots$ are independent and identically distributed random variables that are observed in sequence up to some random time N then

$$E\left[\sum_{i=1}^{N} X_i\right] = E[N]E[X]$$

More precisely let $X_1, X_2, \ldots$ denote a sequence of independent random variables and consider the following definition.

Definition 11.1 An integer-valued random variable N is said to be a *stopping time* for the sequence $X_1, X_2, \ldots$ if the event $\{N = n\}$ is independent of $X_{n+1}, X_{n+2}, \ldots$ for all $n = 1, 2, \ldots$.

Intuitively, we observe the X_ns in sequential order and N denotes the number observed before stopping. If $N = n$, then we have stopped after observing $X_1, \ldots, X_n$ and before observing $X_{n+1}, X_{n+2}, \ldots$ for all $n = 1, 2, \ldots$.

Example 11.6 Let $X_n, n = 1, 2, \ldots$, be independent and such that

$$P\{X_n = 0\} = P\{X_n = 1\} = \tfrac{1}{2}, \quad n = 1, 2, \ldots$$

If we let

$$N = \min\{n: X_1 + \cdots + X_n = 10\}$$

then N is a stopping time. We may regard N as being the stopping time of an experiment that successively flips a fair coin and then stops when the number of heads reaches 10. ■

Proposition 11.4 (Wald's Equation) If $X_1, X_2, \ldots$ are independent and identically distributed random variables having finite expectations, and if N is a stopping time for $X_1, X_2, \ldots$ such that $E[N] < \infty$, then

$$E\left[\sum_1^N X_n\right] = E[N]E[X]$$

Proof. Letting

$$I_n = \begin{cases} 1, & \text{if } N \geqslant n \\ 0, & \text{if } N < n \end{cases}$$

we have

$$\sum_{n=1}^{N} X_n = \sum_{n=1}^{\infty} X_n I_n$$

Hence,

$$E\left[\sum_{n=1}^{N} X_n\right] = E\left[\sum_{n=1}^{\infty} X_n I_n\right] = \sum_{n=1}^{\infty} E[X_n I_n] \tag{11.3}$$

However, $I_n = 1$ if and only if we have not stopped after successively observing $X_1, \ldots, X_{n-1}$. Therefore, I_n is determined by $X_1, \ldots, X_{n-1}$ and is thus

independent of X_n. From Equation (11.3) we thus obtain

$$\begin{aligned} E\left[\sum_{n=1}^{N} X_n\right] &= \sum_{n=1}^{\infty} E[X_n]E[I_n] \\ &= E[X]\sum_{n=1}^{\infty} E[I_n] \\ &= E[X]E\left[\sum_{n=1}^{\infty} I_n\right] \\ &= E[X]E[N] \end{aligned}$$ ■

Returning to the hazard rate method, we have

$$S = \sum_{i=1}^{N} X_i$$

As $N = \min\{n: U_n \leqslant \lambda\left(\sum_1^n X_i\right)/\lambda\}$ it follows that the event that $N = n$ is independent of $X_{n+1}, X_{n+2}, \ldots$. Hence, by Wald's equation,

$$\begin{aligned} E[S] &= E[N]E[X_i] \\ &= \frac{E[N]}{\lambda} \end{aligned}$$

or

$$E[N] = \lambda E[S]$$

where $E[S]$ is the mean of the desired random variable.

11.3 Special Techniques for Simulating Continuous Random Variables

Special techniques have been devised to simulate from most of the common continuous distributions. We now present certain of these.

11.3.1 The Normal Distribution

Let X and Y denote independent standard normal random variables and thus have the joint density function

$$f(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}, \quad -\infty < x < \infty, -\infty < y < \infty$$

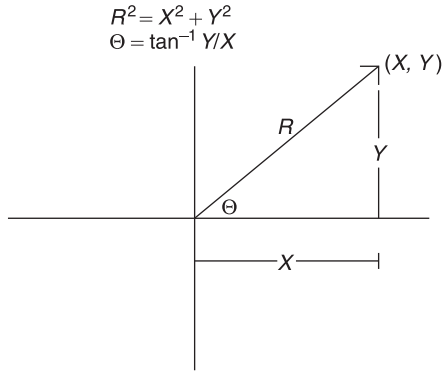

Figure 11.1

Consider now the polar coordinates of the point (X, Y). As shown in Figure 11.1,

$$R^2 = X^2 + Y^2,$$
$$\Theta = \tan^{-1} Y/X$$

To obtain the joint density of R^2 and Θ, consider the transformation

$$d = x^2 + y^2, \quad \theta = \tan^{-1} y/x$$

The Jacobian of this transformation is

$$J = \begin{vmatrix} \frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} \end{vmatrix} = \begin{vmatrix} 2x & 2y \\ \frac{1}{1 + y^2/x^2}\left(\frac{-y}{x^2}\right) & \frac{1}{1 + y^2/x^2}\left(\frac{1}{x}\right) \end{vmatrix}$$
$$= 2\begin{vmatrix} x & y \\ -\frac{y}{x^2 + y^2} & \frac{x}{x^2 + y^2} \end{vmatrix} = 2$$

Hence, from Section 2.5.3 the joint density of R^2 and Θ is given by

$$f_{R^2,\Theta}(d, \theta) = \frac{1}{2\pi} e^{-d/2} \frac{1}{2}$$
$$= \frac{1}{2} e^{-d/2} \frac{1}{2\pi}, \quad 0 < d < \infty, 0 < \theta < 2\pi$$

Thus, we can conclude that R^2 and Θ are independent with R^2 having an exponential distribution with rate $\frac{1}{2}$ and Θ being uniform on $(0, 2\pi)$.

Let us now go in reverse from the polar to the rectangular coordinates. From the preceding if we start with W, an exponential random variable with rate $\frac{1}{2}$ (W plays the role of R^2) and with V, independent of W and uniformly distributed over $(0, 2\pi)$ (V plays the role of Θ) then $X = \sqrt{W}\cos V$, $Y = \sqrt{W}\sin V$ will be independent standard normals. Hence, using the results of Example 11.3 we see that if U_1 and U_2 are independent uniform $(0, 1)$ random numbers, then

$$\begin{aligned} X &= (-2\log\ U_1)^{1/2}\cos(2\pi U_2), \\ Y &= (-2\log\ U_1)^{1/2}\sin(2\pi U_2) \end{aligned} \tag{11.4}$$

are independent standard normal random variables.

Remark The fact that $X^2 + Y^2$ has an exponential distribution with rate $\frac{1}{2}$ is quite interesting for, by the definition of the chi-square distribution, $X^2 + Y^2$ has a chi-squared distribution with two degrees of freedom. Hence, these two distributions are identical.

The preceding approach to generating standard normal random variables is called the *Box–Muller approach*. Its efficiency suffers somewhat from its need to compute the preceding sine and cosine values. There is, however, a way to get around this potentially time-consuming difficulty. To begin, note that if U is uniform on $(0, 1)$, then $2U$ is uniform on $(0, 2)$, and so $2U - 1$ is uniform on $(-1, 1)$. Thus, if we generate random numbers U_1 and U_2 and set

$$\begin{aligned} V_1 &= 2U_1 - 1, \\ V_2 &= 2U_2 - 1 \end{aligned}$$

then (V_1, V_2) is uniformly distributed in the square of area 4 centered at (0, 0) (see Figure 11.2).

Suppose now that we continually generate such pairs (V_1, V_2) until we obtain one that is contained in the circle of radius 1 centered at $(0, 0)$—that is, until (V_1, V_2) is such that $V_1^2 + V_2^2 \leqslant 1$. It now follows that such a pair (V_1, V_2) is uniformly distributed in the circle. If we let $\bar{R}, \bar{\Theta}$ denote the polar coordinates of this pair, then it is easy to verify that $\bar{R}$ and $\bar{\Theta}$ are independent, with $\bar{R}^2$ being uniformly distributed on $(0, 1)$, and $\bar{\Theta}$ uniformly distributed on $(0, 2\pi)$.

Since

$$\begin{aligned} \sin\bar{\Theta} = V_2/\bar{R} &= \frac{V_2}{\sqrt{V_1^2 + V_2^2}}, \\ \cos\bar{\Theta} = V_1/\bar{R} &= \frac{V_1}{\sqrt{V_1^2 + V_2^2}} \end{aligned}$$

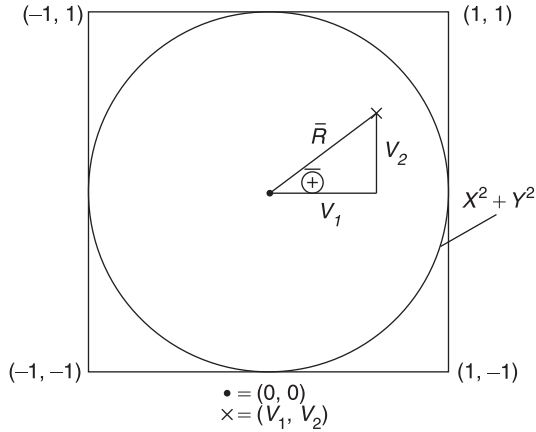

Figure 11.2

it follows from Equation (11.4) that we can generate independent standard normals X and Y by generating another random number U and setting

$$X = (-2 \log\ U)^{1/2} V_1/\bar{R},$$

$$Y = (-2 \log\ U)^{1/2} V_2/\bar{R}$$

In fact, since (conditional on $V_1^2 + V_2^2 \leqslant 1$) $\bar{R}^2$ is uniform on $(0, 1)$ and is independent of $\bar{\Theta}$, we can use it instead of generating a new random number U; thus showing that

$$X = \left(-2 \log \bar{R}^2\right)^{1/2} V_1/\bar{R} = \sqrt{\frac{-2 \log S}{S}}\, V_1,$$

$$Y = \left(-2 \log \bar{R}^2\right)^{1/2} V_2/\bar{R} = \sqrt{\frac{-2 \log S}{S}}\, V_2$$

are independent standard normals, where

$$S = \bar{R}^2 = V_1^2 + V_2^2$$

Summing up, we thus have the following approach to generating a pair of independent standard normals:

Step 1: Generate random numbers U_1 and U_2.
Step 2: Set $V_1 = 2U_1 - 1,\ V_2 = 2U_2 - 1,\ S = V_1^2 + V_2^2$.
Step 3: If $S > 1$, return to step 1.

Step 4: Return the independent unit normals

$$X = \sqrt{\frac{-2\log S}{S}}V_1, \quad Y = \sqrt{\frac{-2\log S}{S}}V_2$$

The preceding is called the *polar method*. Since the probability that a random point in the square will fall within the circle is equal to $\pi/4$ (the area of the circle divided by the area of the square), it follows that, on average, the polar method will require $4/\pi = 1.273$ iterations of step 1. Hence, it will, on average, require 2.546 random numbers, 1 logarithm, 1 square root, 1 division, and 4.546 multiplications to generate 2 independent standard normals.

11.3.2 The Gamma Distribution

To simulate from a gamma distribution with parameters (n, λ), where n is an integer, we use the fact that the sum of n independent exponential random variables each having rate λ has this distribution. Hence, if $U_1, \ldots, U_n$ are independent uniform $(0, 1)$ random variables,

$$X = \frac{1}{\lambda}\sum_{i=1}^{n} \log\ U_i = -\frac{1}{\lambda}\ \log\left(\prod_{i=1}^{n}\ U_i\right)$$

has the desired distribution.

When n is large, there are other techniques available that do not require so many random numbers. One possibility is to use the rejection procedure with $g(x)$ being taken as the density of an exponential random variable with mean n/λ (as this is the mean of the gamma). It can be shown that for large n the average number of iterations needed by the rejection algorithm is $e[(n-1)/2\pi]^{1/2}$. In addition, if we wanted to generate a series of gammas, then, just as in Example 11.4, we can arrange things so that upon acceptance we obtain not only a gamma random variable but also, for free, an exponential random variable that can then be used in obtaining the next gamma (see Exercise 8).

11.3.3 The Chi-Squared Distribution

The chi-squared distribution with n degrees of freedom is the distribution of $\chi_n^2 = Z_1^2 + \cdots + Z_n^2$ where $Z_i, i = 1, \ldots, n$ are independent standard normals. Using the fact noted in the remark at the end of Section 3.1 we see that $Z_1^2 + Z_2^2$ has an exponential distribution with rate $\frac{1}{2}$. Hence, when n is even—say, $n = 2k$—χ_{2k}^2 has a gamma distribution with parameters $\left(k, \frac{1}{2}\right)$. Hence, $-2\log\left(\prod_{i=1}^{k} U_i\right)$ has a chi-squared distribution with $2k$ degrees of freedom. We can simulate a chi-squared random variable with $2k + 1$ degrees of freedom by first simulating a standard normal random variable Z and then adding Z^2 to the

preceding. That is,

$$\chi^2_{2k+1} = Z^2 - 2\log\left(\prod_{i=1}^{k} U_i\right)$$

where $Z, U_1, \ldots, U_n$ are independent with Z being a standard normal and the others being uniform $(0, 1)$ random variables.

11.3.4 The Beta (n, m) Distribution

The random variable X is said to have a beta distribution with parameters n, m if its density is given by

$$f(x) = \frac{(n+m-1)!}{(n-1)!(m-1)!}x^{n-1}(1-x)^{m-1}, \quad 0 < x < 1$$

One approach to simulating from the preceding distribution is to let $U_1, \ldots, U_{n+m-1}$ be independent uniform $(0, 1)$ random variables and consider the nth smallest value of this set—call it $U_{(n)}$. Now $U_{(n)}$ will equal x if, of the $n + m - 1$ variables,

(i) $n - 1$ are smaller than x,
(ii) one equals x,
(iii) $m - 1$ are greater than x.

Hence, if the $n + m - 1$ uniform random variables are partitioned into three subsets of sizes $n - 1$, 1, and $m - 1$ the probability (density) that each of the variables in the first set is less than x, the variable in the second set equals x, and all the variables in the third set are greater than x is given by

$$(P\{U < x\})^{n-1} f_U(x) (P\{U > x\})^{m-1} = x^{n-1}(1-x)^{m-1}$$

Hence, as there are $(n + m - 1)!/(n - 1)!(m - 1)!$ possible partitions, it follows that $U_{(n)}$ is beta with parameters (n, m).

Thus, one way to simulate from the beta distribution is to find the nth smallest of a set of $n + m - 1$ random numbers. However, when n and m are large, this procedure is not particularly efficient.

For another approach consider a Poisson process with rate 1, and recall that given S_{n+m}, the time of the $(n+m)$th event, the set of the first $n+m-1$ event times is distributed independently and uniformly on $(0, S_{n+m})$. Hence, given S_{n+m}, the nth smallest of the first $n + m-1$ event times—that is, S_n—is distributed as the nth smallest of a set of $n + m-1$ uniform $(0, S_{n+m})$ random variables. But

from the preceding we can thus conclude that S_n/S_{n+m} has a beta distribution with parameters (n, m). Therefore, if $U_1, \ldots, U_{n+m}$ are random numbers,

$$\frac{-\log \prod_{i=1}^{n} U_i}{-\log \prod_{i=1}^{m+n} U_i} \text{ is beta with parameters } (n, m)$$

By writing the preceding as

$$\frac{-\log \prod_{i=1}^{n} U_i}{-\log \prod_{1}^{n} U_i - \log \prod_{n+1}^{n+m} U_i}$$

we see that it has the same distribution as $X/(X + Y)$ where X and Y are independent gamma random variables with respective parameters $(n, 1)$ and $(m, 1)$. Hence, when n and m are large, we can efficiently simulate a beta by first simulating two gamma random variables.

11.3.5 The Exponential Distribution—The Von Neumann Algorithm

As we have seen, an exponential random variable with rate 1 can be simulated by computing the negative of the logarithm of a random number. Most computer programs for computing a logarithm, however, involve a power series expansion, and so it might be useful to have at hand a second method that is computationally easier. We now present such a method due to Von Neumann.

To begin let $U_1, U_2, \ldots$ be independent uniform $(0, 1)$ random variables and define $N, N \geqslant 2$, by

$$N = \min\{n: U_1 \geqslant U_2 \geqslant \cdots \geqslant U_{n-1} < U_n\}$$

That is, N is the index of the first random number that is greater than its predecessor. Let us now compute the joint distribution of N and U_1.

$$\begin{aligned} P\{N > n, U_1 \leqslant y\} &= \int_0^1 P\{N > n, U_1 \leqslant y | U_1 = x\}\, dx \\ &= \int_0^y P\{N > n | U_1 = x\}\, dx \end{aligned}$$

Now, given that $U_1 = x, N$ will be greater than n if $x \geqslant U_2 \geqslant \cdots \geqslant U_n$ or, equivalently, if

(a) $U_i \leqslant x, \quad i = 2, \ldots, n$

and

(b) $U_2 \geqslant \cdots \geqslant U_n$

Now, (a) has probability x^{n-1} of occurring and given (a), since all of the $(n-1)!$ possible rankings of $U_2, \ldots, U_n$ are equally likely, (b) has probability $1/(n-1)!$ of occurring. Hence,

$$P\{N > n | U_1 = x\} = \frac{x^{n-1}}{(n-1)!}$$

and so

$$P\{N > n, U_1 \leqslant y\} = \int_0^y \frac{x^{n-1}}{(n-1)!}\, dx = \frac{y^n}{n!}$$

which yields

$$\begin{aligned} P\{N = n, U_1 \leqslant y\} &= P\{N > n-1, U_1 \leqslant y\} - P\{N > n, U_1 \leqslant y\} \\ &= \frac{y^{n-1}}{(n-1)!} - \frac{y^n}{n!} \end{aligned}$$

Upon summing over all the even integers, we see that

$$\begin{aligned} P\{N \text{ is even}, \ U_1 \leqslant y\} &= y - \frac{y^2}{2!} + \frac{y^3}{3!} - \frac{y^4}{4!} - \cdots \\ &= 1 - e^{-y} \end{aligned} \tag{11.5}$$

We are now ready for the following algorithm for generating an exponential random variable with rate 1.

Step 1: Generate uniform random numbers $U_1, U_2, \ldots$ stopping at $N = \min\{n\colon U_1 \geqslant \cdots \geqslant U_{n-1} < U_n\}$.

Step 2: If N is even accept that run, and go to step 3. If N is odd reject the run, and return to step 1.

Step 3: Set X equal to the number of failed runs plus the first random number in the successful run.

To show that X is exponential with rate 1, first note that the probability of a successful run is, from Equation (11.5) with $y = 1$,

$$P\{N \text{ is even}\} = 1 - e^{-1}$$

Now, in order for X to exceed x, the first $[x]$ runs must all be unsuccessful and the next run must either be unsuccessful or be successful but have $U_1 > x - [x]$ (where $[x]$ is the largest integer not exceeding x). As

$$\begin{aligned} P\{N \text{ even}, \ U_1 > y\} &= P\{N \text{ even}\} - P\{N \text{ even}, \ U_1 \leqslant y\} \\ &= 1 - e^{-1} - (1 - e^{-y}) = e^{-y} - e^{-1} \end{aligned}$$

we see that

$$P\{X > x\} = e^{-[x]}\left[e^{-1} + e^{-(x-[x])} - e^{-1}\right] = e^{-x}$$

which yields the result.

Let T denote the number of trials needed to generate a successful run. As each trial is a success with probability $1 - e^{-1}$ it follows that T is geometric with mean $1/(1 - e^{-1})$. If we let N_i denote the number of uniform random variables used on the ith run, $i \geqslant 1$, then T (being the first run i for which N_i is even) is a stopping time for this sequence. Hence, by Wald's equation, the mean number of uniform random variables needed by this algorithm is given by

$$E\left[\sum_{i=1}^{T} N_i\right] = E[N]E[T]$$

Now,

$$\begin{aligned} E[N] &= \sum_{n=0}^{\infty} P\{N > n\} \\ &= 1 + \sum_{n=1}^{\infty} P\{U_1 \geqslant \cdots \geqslant U_n\} \\ &= 1 + \sum_{n=1}^{\infty} 1/n! = e \end{aligned}$$

and so

$$E\left[\sum_{i=1}^{T} N_i\right] = \frac{e}{1 - e^{-1}} \approx 4.3$$

Hence, this algorithm, which computationally speaking is quite easy to perform, requires on the average about 4.3 random numbers to execute.

11.4 Simulating from Discrete Distributions

All of the general methods for simulating from continuous distributions have analogs in the discrete case. For instance, if we want to simulate a random variable X having probability mass function

$$P\{X = x_j\} = P_j, \quad j = 1, 2, \ldots, \quad \sum_j P_j = 1$$

we can use the following discrete time analog of the inverse transform technique:

To simulate X for which $P\{X = x_j\} = P_j$
let U be uniformly distributed over $(0, 1)$, and set

$$X = \begin{cases} x_1, & \text{if } U < P_1 \\ x_2, & \text{if } P_1 < U < P_1 + P_2 \\ \vdots & \\ x_j, & \text{if } \sum_1^{j-1} P_i < U < \sum_i^{j} P_i \\ \vdots & \end{cases}$$

As,

$$P\{X = x_j\} = P\left\{\sum_1^{j-1} P_i < U < \sum_1^{j} P_i\right\} = P_j$$

we see that X has the desired distribution.

Example 11.7 (The Geometric Distribution) Suppose we want to simulate X such that

$$P\{X = i\} = p(1-p)^{i-1}, \quad i \geqslant 1$$

As

$$\sum_{i=1}^{j-1} P\{X = i\} = 1 - P\{X > j-1\} = 1 - (1-p)^{j-1}$$

we can simulate such a random variable by generating a random number U and then setting X equal to that value j for which

$$1 - (1-p)^{j-1} < U < 1 - (1-p)^j$$

or, equivalently, for which

$$(1-p)^j < 1 - U < (1-p)^{j-1}$$

As $1 - U$ has the same distribution as U, we can thus define X by

$$X = \min\{j\colon (1-p)^j < U\} = \min\left\{j\colon j > \frac{\log U}{\log(1-p)}\right\}$$

$$= 1 + \left[\frac{\log U}{\log(1-p)}\right] \qquad \blacksquare$$

As in the continuous case, special simulation techniques have been developed for the more common discrete distributions. We now present certain of these.

Example 11.8 (Simulating a Binomial Random Variable) A binomial (n, p) random variable can be most easily simulated by recalling that it can be expressed as the sum of n independent Bernoulli random variables. That is, if $U_1, \ldots, U_n$ are independent uniform $(0, 1)$ variables, then letting

$$X_i = \begin{cases} 1, & \text{if } U_i < p \\ 0, & \text{otherwise} \end{cases}$$

it follows that $X \equiv \sum_{i=1}^{n} X_i$ is a binomial random variable with parameters n and p.

One difficulty with this procedure is that it requires the generation of n random numbers. To show how to reduce the number of random numbers needed, note first that this procedure does not use the actual value of a random number U but only whether or not it exceeds p. Using this and the result that the conditional distribution of U given that $U < p$ is uniform on $(0, p)$ and the conditional distribution of U given that $U > p$ is uniform on $(p, 1)$, we now show how we can simulate a binomial (n, p) random variable using only a single random number:

Step 1: Let $\alpha = 1/p, \beta = 1/(1-p)$.
Step 2: Set $k = 0$.
Step 3: Generate a uniform random number U.
Step 4: If $k = n$ stop. Otherwise reset k to equal $k + 1$.
Step 5: If $U \leqslant p$ set $X_k = 1$ and reset U to equal αU. If $U > p$ set $X_k = 0$ and reset U to equal $\beta(U - p)$. Return to step 4.

This procedure generates $X_1, \ldots, X_n$ and $X = \sum_{i=1}^{n} X_i$ is the desired random variable. It works by noting whether $U_k \leqslant p$ or $U_k > p$; in the former case it takes U_{k+1} to equal U_k/p, and in the latter case it takes U_{k+1} to equal $(U_k - p)/(1-p)$.[†] ■

Example 11.9 (Simulating a Poisson Random Variable) To simulate a Poisson random variable with mean λ, generate independent uniform $(0, 1)$ random variables $U_1, U_2, \ldots$ stopping at

$$N + 1 = \min\left\{n: \prod_{i=1}^{n} U_i < e^{-\lambda}\right\}$$

The random variable N has the desired distribution, which can be seen by noting that

$$N = \max\left\{n : \sum_{i=1}^{n} -\log U_i < \lambda\right\}$$

[†]Because of computer round-off errors, a single random number should not be continuously used when n is large.

But $-\log U_i$ is exponential with rate 1, and so if we interpret $-\log U_i$, $i \geqslant 1$, as the interarrival times of a Poisson process having rate 1, we see that $N = N(\lambda)$ would equal the number of events by time λ. Hence N is Poisson with mean λ.

When λ is large we can reduce the amount of computation in the preceding simulation of $N(\lambda)$, the number of events by time λ of a Poisson process having rate 1, by first choosing an integer m and simulating S_m, the time of the mth event of the Poisson process, and then simulating $N(\lambda)$ according to the conditional distribution of $N(\lambda)$ given S_m. Now the conditional distribution of $N(\lambda)$ given S_m is as follows:

$$N(\lambda)|S_m = s \sim m + \text{Poisson}(\lambda - s), \qquad \text{if } s < \lambda$$

$$N(\lambda)|S_m = s \sim \text{Binomial}\left(m - 1, \frac{\lambda}{s}\right), \quad \text{if } s > \lambda$$

where $\sim$ means "has the distribution of." This follows since if the mth event occurs at time s, where $s < \lambda$, then the number of events by time λ is m plus the number of events in (s, λ). On the other hand given that $S_m = s$ the set of times at which the first $m - 1$ events occur has the same distribution as a set of $m - 1$ uniform $(0, s)$ random variables (see Section 5.3.5). Hence, when $\lambda < s$, the number of these that occur by time λ is binomial with parameters $m - 1$ and λ/s. Hence, we can simulate $N(\lambda)$ by first simulating S_m and then simulating, either $P(\lambda - S_m)$, a Poisson random variable with mean $\lambda - S_m$, when $S_m < \lambda$, or simulating $\text{Bin}(m - 1, \lambda/S_m)$, a binomial random variable with parameters $m - 1$ and λ/S_m, when $S_m > \lambda$; and then setting

$$N(\lambda) = \begin{cases} m + P(\lambda - S_m), & \text{if } S_m < \lambda \\ \text{Bin}(m - 1, \lambda/S_m), & \text{if } S_m > \lambda \end{cases}$$

In the preceding it has been found computationally effective to let m be approximately $\frac{7}{8}\lambda$. Of course, S_m is simulated by simulating from a gamma (m, λ) distribution via an approach that is computationally fast when m is large (see Section 11.3.3). ■

There are also rejection and hazard rate methods for discrete distributions but we leave their development as exercises. However, there is a technique available for simulating finite discrete random variables—called the *alias method*—which, though requiring some setup time, is very fast to implement.

11.4.1 The Alias Method

In what follows, the quantities $\mathbf{P}$, $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, $k \leqslant n - 1$ will represent probability mass functions on the integers $1, 2, \ldots, n$—that is, they will be n-vectors of nonnegative numbers summing to 1. In addition, the vector $\mathbf{P}^{(k)}$ will have at most k nonzero components, and each of the $\mathbf{Q}^{(k)}$ will have at most two nonzero components. We show that any probability mass function $\mathbf{P}$ can be represented as

an equally weighted mixture of $n-1$ probability mass functions $\mathbf{Q}$ (each having at most two nonzero components). That is, we show that for suitably defined $\mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(n-1)}, \mathbf{P}$ can be expressed as

$$\mathbf{P} = \frac{1}{n-1} \sum_{k=1}^{n-1} \mathbf{Q}^{(k)} \tag{11.6}$$

As a prelude to presenting the method for obtaining this representation, we will need the following simple lemma whose proof is left as an exercise.

Lemma 11.5 Let $\mathbf{P} = \{P_i,\ i = 1, \ldots, n\}$ denote a probability mass function, then

(a) there exists an i, $1 \leqslant i \leqslant n$, such that $P_i < 1/(n-1)$, and
(b) for this i, there exists a j, $j \neq i$, such that $P_i + P_j \geqslant 1/(n-1)$.

Before presenting the general technique for obtaining the representation of Equation (11.6), let us illustrate it by an example.

Example 11.10 Consider the three-point distribution $\mathbf{P}$ with $P_1 = \frac{7}{16}$, $P_2 = \frac{1}{2}$, $P_3 = \frac{1}{16}$. We start by choosing i and j such that they satisfy the conditions of Lemma 11.5. As $P_3 < \frac{1}{2}$ and $P_3 + P_2 > \frac{1}{2}$, we can work with $i = 3$ and $j = 2$. We will now define a two-point mass function $\mathbf{Q}^{(1)}$ putting all of its weight on 3 and 2 and such that $\mathbf{P}$ will be expressible as an equally weighted mixture between $\mathbf{Q}^{(1)}$ and a second two-point mass function $\mathbf{Q}^{(2)}$. Secondly, all of the mass of point 3 will be contained in $\mathbf{Q}^{(1)}$. As we will have

$$P_j = \frac{1}{2}\left(Q_j^{(1)} + Q_j^{(2)}\right), \quad j = 1, 2, 3 \tag{11.7}$$

and, by the preceding, $Q_3^{(2)}$ is supposed to equal 0, we must therefore take

$$Q_3^{(1)} = 2P_3 = \frac{1}{8}, \quad Q_2^{(1)} = 1 - Q_3^{(1)} = \frac{7}{8}, \quad Q_1^{(1)} = 0$$

To satisfy Equation (11.7), we must then set

$$Q_3^{(2)} = 0, \quad Q_2^{(2)} = 2P_2 - \frac{7}{8} = \frac{1}{8}, \quad Q_1^{(2)} = 2P_1 = \frac{7}{8}$$

Hence, we have the desired representation in this case. Suppose now that the original distribution was the following four-point mass function:

$$P_1 = \frac{7}{16}, \quad P_2 = \frac{1}{4}, \quad P_3 = \frac{1}{8}, \quad P_4 = \frac{3}{16}$$

Now, $P_3 < \frac{1}{3}$ and $P_3 + P_1 > \frac{1}{3}$. Hence our initial two-point mass function—$\mathbf{Q}^{(1)}$—will concentrate on points 3 and 1 (giving no weights to 2 and 4). As the final representation will give weight $\frac{1}{3}$ to $\mathbf{Q}^{(1)}$ and in addition the other $\mathbf{Q}^{(j)}$, $j = 2, 3$, will not give any mass to the value 3, we must have

$$\frac{1}{3}Q_3^{(1)} = P_3 = \frac{1}{8}$$

Hence,

$$Q_3^{(1)} = \frac{3}{8}, \quad Q_1^{(1)} = 1 - \frac{3}{8} = \frac{5}{8}$$

Also, we can write

$$\mathbf{P} = \frac{1}{3}\mathbf{Q}^{(1)} + \frac{2}{3}\mathbf{P}^{(3)}$$

where $\mathbf{P}^{(3)}$, to satisfy the preceding, must be the vector

$$\mathbf{P}_1^{(3)} = \frac{3}{2}\left(P_1 - \frac{1}{3}Q_1^{(1)}\right) = \frac{1}{3}\frac{1}{2},$$

$$\mathbf{P}_2^{(3)} = \frac{3}{2}P_2 = \frac{3}{8},$$

$$\mathbf{P}_3^{(3)} = 0,$$

$$\mathbf{P}_4^{(3)} = \frac{3}{2}P_4 = \frac{9}{32}$$

Note that $\mathbf{P}^{(3)}$ gives no mass to the value 3. We can now express the mass function $\mathbf{P}^{(3)}$ as an equally weighted mixture of two-point mass functions $\mathbf{Q}^{(2)}$ and $\mathbf{Q}^{(3)}$, and we will end up with

$$\begin{aligned}\mathbf{P} &= \frac{1}{3}\mathbf{Q}^{(1)} + \frac{2}{3}\left(\frac{1}{2}\mathbf{Q}^{(2)} + \frac{1}{2}\mathbf{Q}^{(3)}\right)\\ &= \frac{1}{3}\left(\mathbf{Q}^{(1)} + \mathbf{Q}^{(2)} + \mathbf{Q}^{(3)}\right)\end{aligned}$$

(We leave it as an exercise for you to fill in the details.) ■

The preceding example outlines the following general procedure for writing the n-point mass function $\mathbf{P}$ in the form of Equation (11.6) where each of the $\mathbf{Q}^{(i)}$ are mass functions giving all their mass to at most two points. To start, we choose i and j satisfying the conditions of Lemma 11.5. We now define the mass function $\mathbf{Q}^{(1)}$ concentrating on the points i and j and which will contain all of the

mass for point i by noting that, in the representation of Equation (11.6), $Q_i^{(k)} = 0$ for $k = 2, \ldots, n-1$, implying that

$$Q_i^{(1)} = (n-1)P_i, \quad \text{and so} \quad Q_j^{(1)} = 1 - (n-1)P_i$$

Writing

$$\mathbf{P} = \frac{1}{n-1}\mathbf{Q}^{(1)} + \frac{n-2}{n-1}\mathbf{P}^{(n-1)} \tag{11.8}$$

where $\mathbf{P}^{(n-1)}$ represents the remaining mass, we see that

$$\begin{aligned}
P_i^{(n-1)} &= 0, \\
P_j^{(n-1)} &= \frac{n-1}{n-2}\left(P_j - \frac{1}{n-1}Q_j^{(1)}\right) = \frac{n-1}{n-2}\left(P_i + P_j - \frac{1}{n-1}\right), \\
P_k^{(n-1)} &= \frac{n-1}{n-2}P_k, \quad k \neq i \text{ or } j
\end{aligned}$$

That the foregoing is indeed a probability mass function is easily checked—for instance, the nonnegativity of $P_j^{(n-1)}$ follows from the fact that j was chosen so that $P_i + P_j \geqslant 1/(n-1)$.

We may now repeat the foregoing procedure on the $(n-1)$-point probability mass function $\mathbf{P}^{(n-1)}$ to obtain

$$\mathbf{P}^{(n-1)} = \frac{1}{n-2}\mathbf{Q}^{(2)} + \frac{n-3}{n-2}\mathbf{P}^{(n-2)}$$

and thus from Equation (11.8) we have

$$\mathbf{P} = \frac{1}{n-1}\mathbf{Q}^{(1)} + \frac{1}{n-1}\mathbf{Q}^{(2)} + \frac{n-3}{n-1}\mathbf{P}^{(n-2)}$$

We now repeat the procedure on $\mathbf{P}^{(n-2)}$ and so on until we finally obtain

$$\mathbf{P} = \frac{1}{n-1}\left(\mathbf{Q}^{(1)} + \cdots + \mathbf{Q}^{(n-1)}\right)$$

In this way we are able to represent $\mathbf{P}$ as an equally weighted mixture of $n-1$ two-point mass functions. We can now easily simulate from $\mathbf{P}$ by first generating a random integer N equally likely to be either $1, 2, \ldots,$ or $n-1$. If the resulting value N is such that $\mathbf{Q}^{(N)}$ puts positive weight only on the points i_N and j_N, then we can set X equal to i_N if a second random number is less than $Q_{i_N}^{(N)}$ and equal to j_N otherwise. The random variable X will have probability mass function $\mathbf{P}$. That is, we have the following procedure for simulating from $\mathbf{P}$:

Step 1: Generate U_1 and set $N = 1 + [(n-1)U_1]$.
Step 2: Generate U_2 and set

$$X = \begin{cases} i_N, & \text{if } U_2 < Q_{i_N}^{(N)} \\ j_N, & \text{otherwise} \end{cases}$$

Remarks

(i) The preceding is called the alias method because by a renumbering of the **Q**s we can always arrange things so that for each k, $Q_k^{(k)} > 0$. (That is, we can arrange things so that the kth two-point mass function gives positive weight to the value k.) Hence, the procedure calls for simulating N, equally likely to be $1, 2, \ldots,$ or $n-1$, and then if $N = k$ it either accepts k as the value of X, or it accepts for the value of X the "alias" of k (namely, the other value that $\mathbf{Q}^{(k)}$ gives positive weight).

(ii) Actually, it is not necessary to generate a new random number in step 2. Because $N-1$ is the integer part of $(n-1)U_1$, it follows that the remainder $(n-1)U_1 - (N-1)$ is independent of U_1 and is uniformly distributed in $(0, 1)$. Hence, rather than generating a new random number U_2 in step 2, we can use $(n-1)U_1 - (N-1) = (n-1)U_1 - [(n-1)U_1]$.

Example 11.11 Let us return to the problem of Example 11.1, which considers a list of n, not necessarily distinct, items. Each item has a value—$v(i)$ being the value of the item in position i—and we are interested in estimating

$$v = \sum_{i=1}^{n} v(i)/m(i)$$

where $m(i)$ is the number of times the item in position i appears on the list. In words, v is the sum of the values of the (distinct) items on the list.

To estimate v, note that if X is a random variable such that

$$P\{X = i\} = v(i) \Big/ \sum_{1}^{n} v(j), \quad i = 1, \ldots, n$$

then

$$E[1/m(X)] = \frac{\sum_i v(i)/m(i)}{\sum_j v(j)} = v \Big/ \sum_{j=1}^{n} v(j)$$

Hence, we can estimate v by using the alias (or any other) method to generate independent random variables $X_1, \ldots, X_k$ having the same distribution as X and then estimating v by

$$v \approx \frac{1}{k} \sum_{j=1}^{n} v(j) \sum_{i=1}^{k} 1/m(X_i)$$

■

11.5 Stochastic Processes

We can easily simulate a stochastic process by simulating a sequence of random variables. For instance, to simulate the first t time units of a renewal process having interarrival distribution F we can simulate independent random variables $X_1, X_2, \ldots$ having distribution F, stopping at

$$N = \min\{n: X_1 + \cdots + X_n > t\}$$

The $X_i, i \geqslant 1$, represent the interarrival times of the renewal process and so the preceding simulation yields $N-1$ events by time t—the events occurring at times $X_1, X_1 + X_2, \ldots, X_1 + \cdots + X_{N-1}$.

Actually there is another approach for simulating a Poisson process that is quite efficient. Suppose we want to simulate the first t time units of a Poisson process having rate λ. To do so, we can first simulate $N(t)$, the number of events by t, and then use the result that given the value of $N(t)$, the set of $N(t)$ event times is distributed as a set of n independent uniform $(0, t)$ random variables. Hence, we start by simulating $N(t)$, a Poisson random variable with mean λt (by one of the methods given in Example 11.9). Then, if $N(t) = n$, generate a new set of n random numbers—call them $U_1, \ldots, U_n$—and $\{tU_1, \ldots, tU_n\}$ will represent the set of $N(t)$ event times. If we could stop here this would be much more efficient than simulating the exponentially distributed interarrival times. However, we usually desire the event times in increasing order—for instance, for $s < t$,

$$N(s) = \text{number of } U_i : tU_i \leqslant s$$

and so to compute the function $N(s), s \leqslant t$, it is best to first order the values $U_i, i = 1, \ldots, n$ before multiplying by t. However, in doing so you should not use an all-purpose sorting algorithm, such as quick sort (see Example 3.14), but rather one that takes into account that the elements to be sorted come from a uniform $(0, 1)$ population. Such a sorting algorithm of n uniform $(0, 1)$ variables is as follows: Rather than a single list to be sorted of length n we will consider n ordered, or linked, lists of random size. The value U will be put in list i if its value is between $(i-1)/n$ and i/n—that is, U is put in list $[nU] + 1$. The individual lists are then ordered, and the total linkage of all the lists is the desired ordering. As almost all of the n lists will be of relatively small size (for instance, if $n = 1000$ the mean number of lists of size greater than 4 is (using the Poisson approximation to the binomial) approximately equal to $1000(1 - \frac{65}{24}e^{-1}) \simeq 4$) the sorting of individual lists will be quite quick, and so the running time of such an algorithm will be proportional to n (rather than to $n \log n$ as in the best all-purpose sorting algorithms).

An extremely important counting process for modeling purposes is the nonhomogeneous Poisson process, which relaxes the Poisson process assumption of stationary increments. Thus it allows for the possibility that the arrival rate need

not be constant but can vary with time. However, there are few analytical studies that assume a nonhomogeneous Poisson arrival process for the simple reason that such models are not usually mathematically tractable. (For example, there is no known expression for the average customer delay in the single-server exponential service distribution queueing model that assumes a nonhomogeneous arrival process.)‡ Clearly such models are strong candidates for simulation studies.

11.5.1 Simulating a Nonhomogeneous Poisson Process

We now present three methods for simulating a nonhomogeneous Poisson process having intensity function $\lambda(t)$, $0 \leqslant t < \infty$.

Method 1. Sampling a Poisson Process

To simulate the first T time units of a nonhomogeneous Poisson process with intensity function $\lambda(t)$, let λ be such that

$$\lambda(t) \leqslant \lambda \quad \text{for all } t \leqslant T$$

Now, as shown in Chapter 5, such a nonhomogeneous Poisson process can be generated by a random selection of the event times of a Poisson process having rate λ. That is, if an event of a Poisson process with rate λ that occurs at time t is counted (independently of what has transpired previously) with probability $\lambda(t)/\lambda$ then the process of counted events is a nonhomogeneous Poisson process with intensity function $\lambda(t), 0 \leqslant t \leqslant T$. Hence, by simulating a Poisson process and then randomly counting its events, we can generate the desired nonhomogeneous Poisson process. We thus have the following procedure:

Generate independent random variables $X_1, U_1, X_2, U_2, \ldots$ where the X_i are exponential with rate λ and the U_i are random numbers, stopping at

$$N = \min\left\{n: \sum_{i=1}^{n} X_i > T\right\}$$

Now let, for $j = 1, \ldots, N-1$,

$$I_j = \begin{cases} 1, & \text{if } U_j \leqslant \lambda\left(\sum_{i=1}^{j} X_i\right)/\lambda \\ 0, & \text{otherwise} \end{cases}$$

and set

$$J = \{j: I_j = 1\}$$

‡One queueing model that assumes a nonhomogeneous Poisson arrival process and is mathematically tractable is the infinite server model.

Thus, the counting process having events at the set of times $\{\sum_{i=1}^{j} X_i: j \in J\}$ constitutes the desired process.

The foregoing procedure, referred to as the thinning algorithm (because it "thins" the homogeneous Poisson points) will clearly be most efficient, in the sense of having the fewest number of rejected event times, when $\lambda(t)$ is near λ throughout the interval. Thus, an obvious improvement is to break up the interval into subintervals and then use the procedure over each subinterval. That is, determine appropriate values k, $0 < t_1 < t_2 < \cdots < t_k < T$, $\lambda_1, \ldots, \lambda_{k+1}$, such that

$$\lambda(s) \leqslant \lambda_i \quad \text{when } t_{i-1} \leqslant s < t_i, i = 1, \ldots, k+1 \quad (\text{where } t_0 = 0,\ t_{k+1} = T) \tag{11.9}$$

Now simulate the nonhomogeneous Poisson process over the interval (t_{i-1}, t_i) by generating exponential random variables with rate λ_i and accepting the generated event occurring at time s, $s \in (t_{i-1}, t_i)$, with probability $\lambda(s)/\lambda_i$. Because of the memoryless property of the exponential and the fact that the rate of an exponential can be changed upon multiplication by a constant, it follows that there is no loss of efficiency in going from one subinterval to the next. In other words, if we are at $t \in [t_{i-1}, t_i)$ and generate X, an exponential with rate λ_i, which is such that $t + X > t_i$ then we can use $\lambda_i[X - (t_i - t)]/\lambda_{i+1}$ as the next exponential with rate λ_{i+1}. Thus, we have the following algorithm for generating the first t time units of a nonhomogeneous Poisson process with intensity function $\lambda(s)$ when the relations (11.9) are satisfied. In the algorithm, t will represent the present time and I the present interval (that is, $I = i$ when $t_{i-1} \leqslant t < t_i$).

Step 1: $t = 0, I = 1$.

Step 2: Generate an exponential random variable X having rate λ_I.

Step 3: If $t + X < t_I$, reset $t = t + X$, generate a random number U, and accept the event time t if $U \leqslant \lambda(t)/\lambda_I$. Return to step 2.

Step 4: (Step reached if $t + X \geqslant t_I$). Stop if $I = k + 1$. Otherwise, reset $X = (X - t_I + t)\lambda_I/\lambda_{I+1}$. Also reset $t = t_I$ and $I = I + 1$, and go to step 3.

Suppose now that over some subinterval (t_{i-1}, t_i) it follows that $\underline{\lambda}_i > 0$ where

$$\underline{\lambda}_i \equiv \text{infimum } \{\lambda(s): t_{i-1} \leqslant s < t_i\}$$

In such a situation, we should not use the thinning algorithm directly but rather should first simulate a Poisson process with rate $\underline{\lambda}_i$ over the desired interval and then simulate a nonhomogeneous Poisson process with the intensity function $\lambda(s) = \lambda(s) - \underline{\lambda}_i$ when $s \in (t_{i-1}, t_i)$. (The final exponential generated for the Poisson process, which carries one beyond the desired boundary, need not be wasted but can be suitably transformed so as to be reusable.) The superposition (or, merging) of the two processes yields the desired process over the interval. The reason for doing it this way is that it saves the need to generate uniform random

variables for a Poisson distributed number, with mean $\underline{\lambda}_i(t_i - t_{i-1})$ of the event times. For instance, consider the case where

$$\lambda(s) = 10 + s, \quad 0 < s < 1$$

Using the thinning method with $\lambda = 11$ would generate an expected number of 11 events each of which would require a random number to determine whether or not to accept it. On the other hand, to generate a Poisson process with rate 10 and then merge it with a generated nonhomogeneous Poisson process with rate $\lambda(s) = s$, $0 < s < 1$, would yield an equally distributed number of event times but with the expected number needing to be checked to determine acceptance being equal to 1.

Another way to make the simulation of nonhomogeneous Poisson processes more efficient is to make use of superpositions. For instance, consider the process where

$$\lambda(t) = \begin{cases} \exp\{t^2\}, & 0 < t < 1.5 \\ \exp\{2.25\}, & 1.5 < t < 2.5 \\ \exp\{(4-t)^2\}, & 2.5 < t < 4 \end{cases}$$

A plot of this intensity function is given in Figure 11.3. One way of simulating this process up to time 4 is to first generate a Poisson process with rate 1 over this interval; then generate a Poisson process with rate $e - 1$ over this interval, accept all events in (1, 3), and only accept an event at time t that is not contained in (1, 3) with probability $[\lambda(t) - 1]/(e - 1)$; then generate a Poisson process with

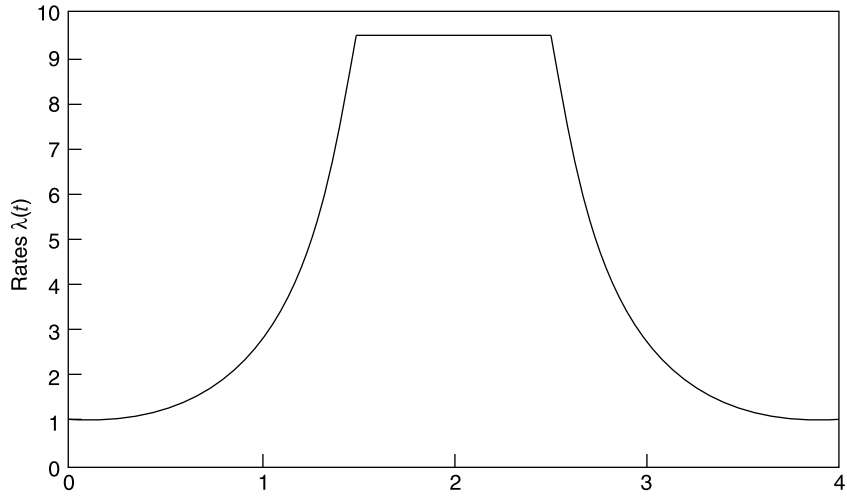

Figure 11.3

rate $e^{2.25} - e$ over the interval (1, 3), accepting all event times between 1.5 and 2.5 and any event time t outside this interval with probability $[\lambda(t) - e]/(e^{2.25} - e)$. The superposition of these processes is the desired nonhomogeneous Poisson process. In other words, what we have done is to break up $\lambda(t)$ into the following nonnegative parts:

$$\lambda(t) = \lambda_1(t) + \lambda_2(t) + \lambda_3(t), \quad 0 < t < 4$$

where

$$\lambda_1(t) \equiv 1,$$

$$\lambda_2(t) = \begin{cases} \lambda(t) - 1, & 0 < t < 1 \\ e - 1, & 1 < t < 3 \\ \lambda(t) - 1, & 3 < t < 4 \end{cases}$$

$$\lambda_3(t) = \begin{cases} \lambda(t) - e, & 1 < t < 1.5 \\ e^{2.25} - e, & 1.5 < t < 2.5 \\ \lambda(t) - e, & 2.5 < t < 3 \\ 0, & 3 < t < 4 \end{cases}$$

and where the thinning algorithm (with a single interval in each case) was used to simulate the constituent nonhomogeneous processes.

Method 2. Conditional Distribution of the Arrival Times

Recall the result for a Poisson process having rate λ that given the number of events by time T the set of event times are independent and identically distributed uniform $(0, T)$ random variables. Now suppose that each of these events is independently counted with a probability that is equal to $\lambda(t)/\lambda$ when the event occurred at time t. Hence, given the number of counted events, it follows that the set of times of these counted events are independent with a common distribution given by $F(s)$, where

$$\begin{aligned} F(s) &= P\{\text{time} \leqslant s | \text{counted}\} \\ &= \frac{P\{\text{time} \leqslant s, \text{counted}\}}{P\{\text{counted}\}} \\ &= \frac{\int_0^T P\{\text{time} \leqslant s, \text{counted} | \text{time} = x\}\, dx/T}{P\{\text{counted}\}} \\ &= \frac{\int_0^s \lambda(x)\, dx}{\int_0^T \lambda(x)\, dx} \end{aligned}$$

The preceding (somewhat heuristic) argument thus shows that given n events of a nonhomogeneous Poisson process by time T the n event times are independent with a common density function

$$f(s) = \frac{\lambda(s)}{m(T)}, \quad 0 < s < T, \quad m(T) = \int_0^T \lambda(s)\, ds \tag{11.10}$$

Since $N(T)$, the number of events by time T, is Poisson distributed with mean $m(T)$, we can simulate the nonhomogeneous Poisson process by first simulating $N(T)$ and then simulating $N(T)$ random variables from the density function of (11.10).

Example 11.12 If $\lambda(s) = cs$, then we can simulate the first T time units of the nonhomogeneous Poisson process by first simulating $N(T)$, a Poisson random variable having mean $m(T) = \int_0^T cs\, ds = CT^2/2$, and then simulating $N(T)$ random variables having distribution

$$F(s) = \frac{s^2}{T^2}, \quad 0 < s < T$$

Random variables having the preceding distribution either can be simulated by use of the inverse transform method (since $F^{-1}(U) = T\sqrt{U}$) or by noting that F is the distribution function of $\max(TU_1, TU_2)$ when U_1 and U_2 are independent random numbers. ■

If the distribution function specified by Equation (11.10) is not easily invertible, we can always simulate from (11.10) by using the rejection method where we either accept or reject simulated values of uniform $(0, T)$ random variables. That is, let $h(s) = 1/T, 0 < s < T$. Then

$$\frac{f(s)}{h(s)} = \frac{T\lambda(s)}{m(T)} \leqslant \frac{\lambda T}{m(T)} \equiv C$$

where λ is a bound on $\lambda(s)$, $0 \leqslant s \leqslant T$. Hence, the rejection method is to generate random numbers U_1 and U_2 then accept TU_1 if

$$U_2 \leqslant \frac{f(TU_1)}{Ch(TU_1)}$$

or, equivalently, if

$$U_2 \leqslant \frac{\lambda(TU_1)}{\lambda}$$

Method 3. Simulating the Event Times

The third method we shall present for simulating a nonhomogeneous Poisson process having intensity function $\lambda(t), t \geqslant 0$ is probably the most basic approach—namely, to simulate the successive event times. So let $X_1, X_2, \ldots$ denote the event times of such a process. As these random variables are dependent we will use the conditional distribution approach to simulation. Hence, we need the conditional distribution of X_i given $X_1, \ldots, X_{i-1}$.

To start, note that if an event occurs at time x then, independent of what has occurred prior to x, the time until the next event has the distribution F_x given by

$$\begin{aligned}\bar{F}_x(t) &= P\{0 \text{ events in } (x, x+t) | \text{event at } x\} \\ &= P\{0 \text{ events in } (x, x+t)\} \quad \text{by independent increments} \\ &= \exp\left\{-\int_0^t \lambda(x+y)\,dy\right\}\end{aligned}$$

Differentiation yields that the density corresponding to F_x is

$$f_x(t) = \lambda(x+t)\exp\left\{-\int_0^t \lambda(x+y)\,dy\right\}$$

implying that the hazard rate function of F_x is

$$r_x(t) = \frac{f_x(t)}{\bar{F}_x(t)} = \lambda(x+t)$$

We can now simulate the event times $X_1, X_2, \ldots$ by simulating X_1 from F_0; then if the simulated value of X_1 is x_1, simulate X_2 by adding x_1 to a value generated from F_{x_1}, and if this sum is x_2 simulate X_3 by adding x_2 to a value generated from F_{x_2}, and so on. The method used to simulate from these distributions should depend, of course, on the form of these distributions. However, it is interesting to note that if we let λ be such that $\lambda(t) \leqslant \lambda$ and use the hazard rate method to simulate, then we end up with the approach of Method 1 (we leave the verification of this fact as an exercise). Sometimes, however, the distributions F_x can be easily inverted and so the inverse transform method can be applied.

Example 11.13 Suppose that $\lambda(x) = 1/(x+a), x \geqslant 0$. Then

$$\int_0^t \lambda(x+y)\,dy = \log\left(\frac{x+a+t}{x+a}\right)$$

Hence,

$$F_x(t) = 1 - \frac{x+a}{x+a+t} = \frac{t}{x+a+t}$$

and so

$$F_x^{-1}(u) = (x + a)\frac{u}{1 - u}$$

We can, therefore, simulate the successive event times $X_1, X_2, \ldots$ by generating $U_1, U_2, \ldots$ and then setting

$$X_1 = \frac{aU_1}{1 - U_1},$$
$$X_2 = (X_1 + a)\frac{U_2}{1 - U_2} + X_1$$

and, in general,

$$X_j = (X_{j-1} + a)\frac{U_j}{1 - U_j} + X_{j-1}, \quad j \geqslant 2$$ ■

11.5.2 Simulating a Two-Dimensional Poisson Process

A point process consisting of randomly occurring points in the plane is said to be a two-dimensional Poisson process having rate λ if

(a) the number of points in any given region of area A is Poisson distributed with mean λA; and
(b) the numbers of points in disjoint regions are independent.

For a given fixed point **O** in the plane, we now show how to simulate events occurring according to a two-dimensional Poisson process with rate λ in a circular region of radius r centered about **O**. Let $R_i, i \geqslant 1$, denote the distance between **O** and its ith nearest Poisson point, and let $C(a)$ denote the circle of radius a centered at **O**. Then

$$P\{\pi R_1^2 > b\} = P\left\{R_1 > \sqrt{\frac{b}{\pi}}\right\} = P\left\{\text{no points in } C\left(\sqrt{b/\pi}\right)\right\} = e^{-\lambda b}$$

Also, with $C(a_2) - C(a_1)$ denoting the region between $C(a_2)$ and $C(a_1)$:

$$\begin{aligned}
&P\{\pi R_2^2 - \pi R_1^2 > b | R_1 = r\} \\
&\quad = P\left\{R_2 > \sqrt{(b + \pi r^2)/\pi}\,\middle|\, R_1 = r\right\} \\
&\quad = P\left\{\text{no points in } C\left(\sqrt{(b + \pi r^2)/\pi}\right) - C(r) \middle| R_1 = r\right\} \\
&\quad = P\left\{\text{no points in } C\left(\sqrt{(b + \pi r^2)/\pi}\right) - C(r)\right\} \quad \text{by (b)} \\
&\quad = e^{-\lambda b}
\end{aligned}$$

In fact, the same argument can be repeated to obtain the following.

Proposition 11.6 With $R_0 = 0$,

$$\pi R_i^2 - \pi R_{i-1}^2, \quad i \geqslant 1,$$

are independent exponentials with rate λ.

In other words, the amount of area that needs to be traversed to encompass a Poisson point is exponential with rate λ. Since, by symmetry, the respective angles of the Poisson points are independent and uniformly distributed over $(0, 2\pi)$, we thus have the following algorithm for simulating the Poisson process over a circular region of radius r about **O**:

Step 1: Generate independent exponentials with rate 1, $X_1, X_2, \ldots$, stopping at

$$N = \min\left\{n: \frac{X_1 + \cdots + X_n}{\lambda\pi} > r^2\right\}$$

Step 2: If $N = 1$, stop. There are no points in $C(r)$. Otherwise, for $i = 1, \ldots, N-1$, set

$$R_i = \sqrt{(X_1 + \cdots + X_i)/\lambda\pi}$$

Step 3: Generate independent uniform $(0, 1)$ random variables $U_1, \ldots, U_{N-1}$.
Step 4: Return the $N - 1$ Poisson points in $C(r)$ whose polar coordinates are

$$(R_i, 2\pi U_i), \quad i = 1, \ldots, N-1$$

The preceding algorithm requires, on average, $1 + \lambda\pi r^2$ exponentials and an equal number of uniform random numbers. Another approach to simulating points in $C(r)$ is to first simulate N, the number of such points, and then use the fact that, given N, the points are uniformly distributed in $C(r)$. This latter procedure requires the simulation of N, a Poisson random variable with mean $\lambda\pi r^2$; we must then simulate N uniform points on $C(r)$, by simulating R from the distribution $F_R(a) = a^2/r^2$ (see Exercise 25) and θ from uniform $(0, 2\pi)$ and must then sort these N uniform values in increasing order of R. The main advantage of the first procedure is that it eliminates the need to sort.

The preceding algorithm can be thought of as the fanning out of a circle centered at **O** with a radius that expands continuously from 0 to r. The successive radii at which Poisson points are encountered is simulated by noting that the additional area necessary to encompass a Poisson point is always, independent of the past, exponential with rate λ. This technique can be used to simulate the process over noncircular regions. For instance, consider a nonnegative function $g(x)$, and suppose we are interested in simulating the Poisson process in the region between the x-axis and g with x going from 0 to T (see Figure 11.4). To do so we can start at the left-hand end and fan vertically to the right by considering the successive areas $\int_0^a g(x)\,dx$. Now if $X_1 < X_2 < \cdots$ denote the successive projections of the Poisson points on the x-axis, then analogous to Proposition 11.6, it will follow that (with $X_0 = 0$) $\lambda \int_{X_{i-1}}^{X_i} g(x)\,dx, i \geqslant 1$, will be independent exponentials

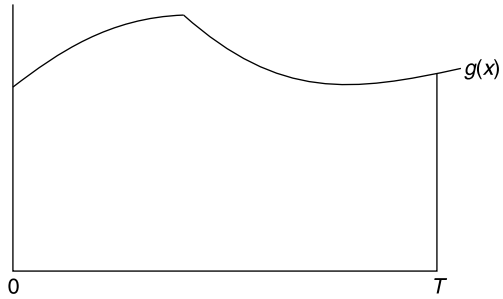

Figure 11.4

with rate 1. Hence, we should simulate $\epsilon_1, \epsilon_2, \ldots$, independent exponentials with rate 1, stopping at

$$N = \min\left\{n: \epsilon_1 + \cdots + \epsilon_n > \lambda \int_0^T g(x)\, dx\right\}$$

and determine $X_1, \ldots, X_{N-1}$ by

$$\begin{aligned}
\lambda \int_0^{X_1} g(x)\, dx &= \epsilon_1, \\
\lambda \int_{X_1}^{X_2} g(x)\, dx &= \epsilon_2, \\
&\vdots \\
\lambda \int_{X_{N-2}}^{X_{N-1}} g(x)\, dx &= \epsilon_{N-1}
\end{aligned}$$

If we now simulate $U_1, \ldots, U_{N-1}$—independent uniform $(0, 1)$ random numbers—then as the projection on the y-axis of the Poisson point whose x-coordinate is X_i is uniform on $(0, g(X_i))$, it follows that the simulated Poisson points in the interval are $(X_i, U_i g(X_i))$, $i = 1, \ldots, N-1$.

Of course, the preceding technique is most useful when g is regular enough so that the foregoing equations can be solved for the X_i. For instance, if $g(x) = y$ (and so the region of interest is a rectangle), then

$$X_i = \frac{\epsilon_1 + \cdots + \epsilon_i}{\lambda y}, \quad i = 1, \ldots, N-1$$

and the Poisson points are

$$(X_i, yU_i), \quad i = 1, \ldots, N-1$$

11.6 Variance Reduction Techniques

Let $X_1, \ldots, X_n$ have a given joint distribution, and suppose we are interested in computing

$$\theta \equiv E[g(X_1, \ldots, X_n)]$$

where g is some specified function. It is often the case that it is not possible to analytically compute the preceding, and when such is the case we can attempt to use simulation to estimate θ. This is done as follows: Generate $X_1^{(1)}, \ldots, X_n^{(1)}$ having the same joint distribution as $X_1, \ldots, X_n$ and set

$$Y_1 = g\big(X_1^{(1)}, \ldots, X_n^{(1)}\big)$$

Now, simulate a second set of random variables (independent of the first set) $X_1^{(2)}, \ldots, X_n^{(2)}$ having the distribution of $X_1, \ldots, X_n$ and set

$$Y_2 = g\big(X_1^{(2)}, \ldots, X_n^{(2)}\big)$$

Continue this until you have generated k (some predetermined number) sets, and so have also computed $Y_1, Y_2, \ldots, Y_k$. Now, $Y_1, \ldots, Y_k$ are independent and identically distributed random variables each having the same distribution of $g(X_1, \ldots, X_n)$. Thus, if we let $\bar{Y}$ denote the average of these k random variables—that is,

$$\bar{Y} = \sum_{i=1}^{k} Y_i/k$$

then

$$E[\bar{Y}] = \theta,$$

$$E\big[(\bar{Y} - \theta)^2\big] = \mathrm{Var}(\bar{Y})$$

Hence, we can use $\bar{Y}$ as an estimate of θ. As the expected square of the difference between $\bar{Y}$ and θ is equal to the variance of $\bar{Y}$, we would like this quantity to be as small as possible. In the preceding situation, $\mathrm{Var}(\bar{Y}) = \mathrm{Var}(Y_i)/k$, which is usually not known in advance but must be estimated from the generated values $Y_1, \ldots, Y_n$. We now present three general techniques for reducing the variance of our estimator.

11.6.1 Use of Antithetic Variables

In the preceding situation, suppose that we have generated Y_1 and Y_2, identically distributed random variables having mean θ. Now,

$$\begin{aligned}\operatorname{Var}\left(\frac{Y_1 + Y_2}{2}\right) &= \frac{1}{4}[\operatorname{Var}(Y_1) + \operatorname{Var}(Y_2) + 2\operatorname{Cov}(Y_1, Y_2)] \\ &= \frac{\operatorname{Var}(Y_1)}{2} + \frac{\operatorname{Cov}(Y_1, Y_2)}{2}\end{aligned}$$

Hence, it would be advantageous (in the sense that the variance would be reduced) if Y_1 and Y_2 rather than being independent were negatively correlated. To see how we could arrange this, let us suppose that the random variables $X_1, \ldots, X_n$ are independent and, in addition, that each is simulated via the inverse transform technique. That is, X_i is simulated from $F_i^{-1}(U_i)$ where U_i is a random number and F_i is the distribution of X_i. Hence, Y_1 can be expressed as

$$Y_1 = g\left(F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n)\right)$$

Now, since $1 - U$ is also uniform over $(0, 1)$ whenever U is a random number (and is negatively correlated with U) it follows that Y_2 defined by

$$Y_2 = g\left(F_1^{-1}(1 - U_1), \ldots, F_n^{-1}(1 - U_n)\right)$$

will have the same distribution as Y_1. Hence, if Y_1 and Y_2 were negatively correlated, then generating Y_2 by this means would lead to a smaller variance than if it were generated by a new set of random numbers. (In addition, there is a computational savings since rather than having to generate n additional random numbers, we need only subtract each of the previous n from 1.) The following theorem will be the key to showing that this technique—known as the use of *antithetic variables*—will lead to a reduction in variance whenever g is a monotone function.

Theorem 11.1 If $X_1, \ldots, X_n$ are independent, then, for any increasing functions f and g of n variables,

$$E[f(\mathbf{X})g(\mathbf{X})] \geqslant E[f(\mathbf{X})]E[g(\mathbf{X})] \tag{11.11}$$

where $\mathbf{X} = (X_1, \ldots, X_n)$.

Proof. The proof is by induction on n. To prove it when $n = 1$, let f and g be increasing functions of a single variable. Then, for any x and y,

$$(f(x) - f(y))(g(x) - g(y)) \geqslant 0$$

since if $x \geqslant y$ ($x \leqslant y$) then both factors are nonnegative (nonpositive). Hence, for any random variables X and Y,

$$(f(X) - f(Y))(g(X) - g(Y)) \geqslant 0$$

implying that

$$E[(f(X) - f(Y))(g(X) - g(Y))] \geqslant 0$$

or, equivalently,

$$E[f(X)g(X)] + E[f(Y)g(Y)] \geqslant E[f(X)g(Y)] + E[f(Y)g(X)]$$

If we suppose that X and Y are independent and identically distributed, as in this case, then

$$E[f(X)g(X)] = E[f(Y)g(Y)],$$
$$E[f(X)g(Y)] = E[f(Y)g(X)] = E[f(X)]E[g(X)]$$

and so we obtain the result when $n = 1$.

So assume that (11.11) holds for $n-1$ variables, and now suppose that $X_1, \ldots, X_n$ are independent and f and g are increasing functions. Then

$$\begin{aligned}
&E[f(\mathbf{X})g(\mathbf{X})|X_n = x_n] \\
&\quad = E[f(X_1, \ldots, X_{n-1}, x_n)g(X_1, \ldots, X_{n-1}, x_n)|X_n = x] \\
&\quad = E[f(X_1, \ldots, X_{n-1}, x_n)g(X_1, \ldots, X_{n-1}, x_n)] \quad \text{by independence} \\
&\quad \geqslant E[f(X_1, \ldots, X_{n-1}, x_n)]E[g(X_1, \ldots, X_{n-1}, x_n)] \\
&\qquad \text{by the induction hypothesis} \\
&\quad = E[f(\mathbf{X})|X_n = x_n]E[g(\mathbf{X})|X_n = x_n]
\end{aligned}$$

Hence,

$$E[f(\mathbf{X})g(\mathbf{X})|X_n] \geqslant E[f(\mathbf{X})|X_n]E[g(\mathbf{X})|X_n]$$

and, upon taking expectations of both sides,

$$\begin{aligned}
E[f(\mathbf{X})g(\mathbf{X})] &\geqslant E\big[E[f(\mathbf{X})|X_n]E[g(\mathbf{X})|X_n]\big] \\
&\geqslant E[f(\mathbf{X})]E[g(\mathbf{X})]
\end{aligned}$$

The last inequality follows because $E[f(\mathbf{X})|X_n]$ and $E[g(\mathbf{X})|X_n]$ are both increasing functions of X_n, and so, by the result for $n = 1$,

$$\begin{aligned}
E\big[E[f(\mathbf{X})|X_n]E[g(\mathbf{X})|X_n]\big] &\geqslant E\big[E[f(\mathbf{X})|X_n]\big]E\big[E[g(\mathbf{X})|X_n]\big] \\
&= E[f(\mathbf{X})]E[g(\mathbf{X})]
\end{aligned}$$

■

Corollary 11.7 If $U_1, \ldots, U_n$ are independent, and k is either an increasing or decreasing function, then

$$\text{Cov}\big(k(U_1, \ldots, U_n), k(1 - U_1, \ldots, 1 - U_n)\big) \leqslant 0$$

Proof. Suppose k is increasing. As $-k(1 - U_1, \ldots, 1 - U_n)$ is increasing in $U_1, \ldots, U_n$, then, from Theorem 11.1,

$$\text{Cov}\big(k(U_1, \ldots, U_n), -k(1 - U_1, \ldots, 1 - U_n)\big) \geqslant 0$$

When k is decreasing just replace k by its negative. ■

Since $F_i^{-1}(U_i)$ is increasing in U_i (as F_i, being a distribution function, is increasing) it follows that $g(F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n))$ is a monotone function of $U_1, \ldots, U_n$ whenever g is monotone. Hence, if g is monotone the antithetic variable approach of twice using each set of random numbers $U_1, \ldots, U_n$ by first computing $g(F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n))$ and then $g(F_1^{-1}(1 - U_1), \ldots, F_n^{-1}(1 - U_n))$ will reduce the variance of the estimate of $E[g(X_1, \ldots, X_n)]$. That is, rather than generating k sets of n random numbers, we should generate $k/2$ sets and use each set twice.

Example 11.14 (Simulating the Reliability Function) Consider a system of n components in which component i, independently of other components, works with probability p_i, $i = 1, \ldots, n$. Letting

$$X_i = \begin{cases} 1, & \text{if component } i \text{ works} \\ 0, & \text{otherwise} \end{cases}$$

suppose there is a monotone structure function ϕ such that

$$\phi(X_1, \ldots, X_n) = \begin{cases} 1, & \text{if the system works under } X_1, \ldots, X_n \\ 0, & \text{otherwise} \end{cases}$$

We are interested in using simulation to estimate

$$r(p_1, \ldots, p_n) \equiv E[\phi(X_1, \ldots, X_n)] = P\{\phi(X_1, \ldots, X_n) = 1\}$$

Now, we can simulate the X_i by generating uniform random numbers $U_1, \ldots, U_n$ and then setting

$$X_i = \begin{cases} 1, & \text{if } U_i < p_i \\ 0, & \text{otherwise} \end{cases}$$

Hence, we see that

$$\phi(X_1, \ldots, X_n) = k(U_1, \ldots, U_n)$$

where k is a decreasing function of $U_1, \ldots, U_n$. Hence,

$$\text{Cov}(k(\mathbf{U}), k(\mathbf{1} - \mathbf{U})) \leqslant 0$$

and so the antithetic variable approach of using $U_1, \ldots, U_n$ to generate both $k(U_1, \ldots, U_n)$ and $k(1 - U_1, \ldots, 1 - U_n)$ results in a smaller variance than if an independent set of random numbers was used to generate the second k. ■

Example 11.15 (Simulating a Queueing System) Consider a given queueing system, let D_i denote the delay in queue of the ith arriving customer, and suppose we are interested in simulating the system so as to estimate

$$\theta = E[D_1 + \cdots + D_n]$$

Let $X_1, \ldots, X_n$ denote the first n interarrival times and $S_1, \ldots, S_n$ the first n service times of this system, and suppose these random variables are all independent. Now in most systems $D_1 + \cdots + D_n$ will be a function of $X_1, \ldots, X_n$, $S_1, \ldots, S_n$—say,

$$D_1 + \cdots + D_n = g(X_1, \ldots, X_n, S_1, \ldots, S_n)$$

Also, g will usually be increasing in S_i and decreasing in $X_i, i = 1, \ldots, n$. If we use the inverse transform method to simulate $X_i, S_i, i = 1, \ldots, n$—say, $X_i = F_i^{-1}(1 - U_i), S_i = G_i^{-1}(\bar{U}_i)$ where $U_1, \ldots, U_n, \bar{U}_1, \ldots, \bar{U}_n$ are independent uniform random numbers—then we may write

$$D_1 + \cdots + D_n = k(U_1, \ldots, U_n, \bar{U}_1, \ldots, \bar{U}_n)$$

where k is increasing in its variates. Hence, the antithetic variable approach will reduce the variance of the estimator of θ. (Thus, we would generate $U_i, \bar{U}_i$, $i = 1, \ldots, n$ and set $X_i = F_i^{-1}(1 - U_i)$ and $Y_i = G_i^{-1}(\bar{U}_i)$ for the first run, and $X_i = F_i^{-1}(U_i)$ and $Y_i = G_i^{-1}(1 - \bar{U}_i)$ for the second.) As all the U_i and $\bar{U}_i$ are independent, however, this is equivalent to setting $X_i = F_i^{-1}(U_i), Y_i = G_i^{-1}(\bar{U}_i)$ in the first run and using $1 - U_i$ for U_i and $1 - \bar{U}_i$ for $\bar{U}_i$ in the second. ■

11.6.2 Variance Reduction by Conditioning

Let us start by recalling (see Proposition 3.1) the conditional variance formula

$$\text{Var}(Y) = E[\text{Var}(Y|Z)] + \text{Var}(E[Y|Z]) \tag{11.12}$$

Now suppose we are interested in estimating $E[g(X_1, \ldots, X_n)]$ by simulating $\mathbf{X} = (X_1, \ldots, X_n)$ and then computing $Y = g(X_1, \ldots, X_n)$. Now, if for some random variable Z we can compute $E[Y|Z]$ then, as $\text{Var}(Y|Z) \geqslant 0$, it follows

from the conditional variance formula that

$$\text{Var}(E[Y|Z]) \leqslant \text{Var}(Y)$$

implying, since $E[E[Y|Z]] = E[Y]$, that $E[Y|Z]$ is a better estimator of $E[Y]$ than is Y.

In many situations, there are a variety of Z_i that can be conditioned on to obtain an improved estimator. Each of these estimators $E[Y|Z_i]$ will have mean $E[Y]$ and smaller variance than does the raw estimator Y. We now show that for any choice of weights $\lambda_i, \lambda_i \geqslant 0, \sum_i \lambda_i = 1$, $\sum_i \lambda_i E[Y|Z_i]$ is also an improvement over Y.

Proposition 11.8 For any $\lambda_i \geqslant 0, \sum_{i=1}^{\infty} \lambda_i = 1$,

(a) $E\left[\sum_i \lambda_i E[Y|Z_i]\right] = E[Y]$,
(b) $\text{Var}\left(\sum_i \lambda_i E[Y|Z_i]\right) \leqslant \text{Var}(Y)$.

Proof. The proof of (a) is immediate. To prove (b), let N denote an integer valued random variable independent of all the other random variables under consideration and such that

$$P\{N = i\} = \lambda_i, \quad i \geqslant 1$$

Applying the conditional variance formula twice yields

$$\begin{aligned}\text{Var}(Y) &\geqslant \text{Var}(E[Y|N, Z_N]) \\ &\geqslant \text{Var}\big(E[E[Y|N, Z_N]|Z_1, \ldots]\big) \\ &= \text{Var} \sum_i \lambda_i E[Y|Z_i]\end{aligned}$$

■

Example 11.16 Consider a queueing system having Poisson arrivals and suppose that any customer arriving when there are already N others in the system is lost. Suppose that we are interested in using simulation to estimate the expected number of lost customers by time t. The raw simulation approach would be to simulate the system up to time t and determine L, the number of lost customers for that run. A better estimate, however, can be obtained by conditioning on the total time in $[0, t]$ that the system is at capacity. Indeed, if we let T denote the time in $[0, t]$ that there are N in the system, then

$$E[L|T] = \lambda T$$

where λ is the Poisson arrival rate. Hence, a better estimate for $E[L]$ than the average value of L over all simulation runs can be obtained by multiplying the average value of T per simulation run by λ. If the arrival process were a nonhomogeneous

Poisson process, then we could improve over the raw estimator L by keeping track of those time periods for which the system is at capacity. If we let $I_1, \ldots, I_C$ denote the time intervals in $[0, t]$ in which there are N in the system, then

$$E[L|I_1, \ldots, I_C] = \sum_{i=1}^{C} \int_{I_i} \lambda(s)\, ds$$

where $\lambda(s)$ is the intensity function of the nonhomogeneous Poisson arrival process. The use of the right side of the preceding would thus lead to a better estimate of $E[L]$ than the raw estimator L. ■

Example 11.17 Suppose that we wanted to estimate the expected sum of the times in the system of the first n customers in a queueing system. That is, if W_i is the time that the ith customer spends in the system, then we are interested in estimating

$$\theta = E\left[\sum_{i=1}^{n} W_i\right]$$

Let Y_i denote the "state of the system" at the moment at which the ith customer arrives. It can be shown[§] that for a wide class of models the estimator $\sum_{i=1}^{n} E[W_i|Y_i]$ has (the same mean and) a smaller variance than the estimator $\sum_{i=1}^{n} W_i$. (It should be noted that whereas it is immediate that $E[W_i|Y_i]$ has smaller variance than W_i, because of the covariance terms involved it is not immediately apparent that $\sum_{i=1}^{n} E[W_i|Y_i]$ has smaller variance than $\sum_{i=1}^{n} W_i$.) For instance, in the model $G/M/1$

$$E[W_i|Y_i] = (N_i + 1)/\mu$$

where N_i is the number in the system encountered by the ith arrival and $1/\mu$ is the mean service time; the result implies that $\sum_{i=1}^{n}(N_i + 1)/\mu$ is a better estimate of the expected total time in the system of the first n customers than is the raw estimator $\sum_{i=1}^{n} W_i$. ■

Example 11.18 (Estimating the Renewal Function by Simulation) Consider a queueing model in which customers arrive daily in accordance with a renewal process having interarrival distribution F. However, suppose that at some fixed time T, for instance 5 P.M., no additional arrivals are permitted and those customers that are still in the system are serviced. At the start of the next and each succeeding day customers again begin to arrive in accordance with the renewal process.

[§] S. M. Ross, "Simulating Average Delay—Variance Reduction by Conditioning," *Probability in the Engineering and Informational Sciences* 2(3), (1988), pp. 309–312.

Suppose we are interested in determining the average time that a customer spends in the system. Upon using the theory of renewal reward processes (with a cycle starting every T time units), it can be shown that

$$\text{average time that a customer spends in the system} = \frac{E[\text{sum of the times in the system of arrivals in } (0, T)]}{m(T)}$$

where $m(T)$ is the expected number of renewals in $(0, T)$.

If we were to use simulation to estimate the preceding quantity, a run would consist of simulating a single day, and as part of a simulation run, we would observe the quantity $N(T)$, the number of arrivals by time T. Since $E[N(T)] = m(T)$, the natural simulation estimator of $m(T)$ would be the average (over all simulated days) value of $N(T)$ obtained. However, $\text{Var}(N(T))$ is, for large T, proportional to T (its asymptotic form being $T\sigma^2/\mu^3$, where σ^2 is the variance and μ the mean of the interarrival distribution F), and so, for large T, the variance of our estimator would be large. A considerable improvement can be obtained by using the analytic formula (see Section 7.3)

$$m(T) = \frac{T}{\mu} - 1 + \frac{E[Y(T)]}{\mu} \tag{11.13}$$

where $Y(T)$ denotes the time from T until the next renewal—that is, it is the excess life at T. Since the variance of $Y(T)$ does not grow with T (indeed, it converges to a finite value provided the moments of F are finite), it follows that for T large, we would do much better by using the simulation to estimate $E[Y(T)]$ and then using Equation (11.13) to estimate $m(T)$.

However, by employing conditioning, we can improve further on our estimate of $m(T)$. To do so, let $A(T)$ denote the age of the renewal process at time T—that is, it is the time at T since the last renewal. Then, rather than using the value of $Y(T)$, we can reduce the variance by considering $E[Y(T)|A(T)]$. Now, knowing that the age at T is equal to x is equivalent to knowing that there was a renewal at time $T - x$ and the next interarrival time X is greater than x. Since the excess at T will equal $X - x$ (see Figure 11.5), it follows that

$$\begin{aligned} E[Y(T)|A(T) = x] &= E[X - x|X > x] \\ &= \int_0^\infty \frac{P\{X - x > t\}}{P\{X > x\}}\, dt \\ &= \int_0^\infty \frac{[1 - F(t + x)]}{1 - F(x)}\, dt \end{aligned}$$

which can be numerically evaluated if necessary.

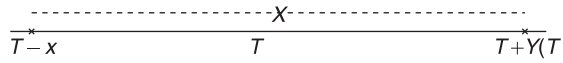

Figure 11.5 $A(T) = x$.

As an illustration of the preceding note that if the renewal process is a Poisson process with rate λ, then the raw simulation estimator $N(T)$ will have variance λT; since $Y(T)$ will be exponential with rate λ, the estimator based on (11.13) will have variance $\lambda^2 \operatorname{Var}\{Y(T)\} = 1$. On the other hand, since $Y(T)$ will be independent of $A(T)$ (and $E[Y(T)|A(T)] = 1/\lambda$), it follows that the variance of the improved estimator $E[Y(T)|A(T)]$ is 0. That is, conditioning on the age at time T yields, in this case, the exact answer. ■

Example 11.19 Consider the $M/G/1$ queueing system where customers arrive in accordance with a Poisson process with rate λ to a single server having service distribution G with mean $E[S]$. Suppose that, for a specified time t_0, the server will take a break at the first time $t \geqslant t_0$ at which the system is empty. That is, if $X(t)$ is the number of customers in the system at time t, then the server will take a break at time

$$T = \min\{t \geqslant t_0 : X(t) = 0\}$$

To efficiently use simulation to estimate $E[T]$, generate the system to time t_0; let R denote the remaining service time of the customer in service at time t_0, and let X_Q equal the number of customers waiting in queue at time t_0. (Note that R is equal to 0 if $X(t_0) = 0$, and $X_Q = (X(t_0) - 1)^+$.) Now, with N equal to the number of customers that arrive in the remaining service time R, it follows that if $N = n$ and $X_Q = n_Q$, then the additional amount of time from $t_0 + R$ until the server can take a break is equal to the amount of time that it takes until the system, starting with $n + n_Q$ customers, becomes empty. Because this is equal to the sum of $n + n_Q$ busy periods, it follows from Section 8.5.3 that

$$E[T|R, N, X_Q] = t_0 + R + (N + X_Q)\frac{E[S]}{1 - \lambda E[S]}$$

Consequently,

$$\begin{aligned} E[T|R, X_Q] &= E\big[E[T|R, N, X_Q]|R, X_Q\big] \\ &= t_0 + R + (E[N|R, X_Q] + X_Q)\frac{E[S]}{1 - \lambda E[S]} \\ &= t_0 + R + (\lambda R + X_Q)\frac{E[S]}{1 - \lambda E[S]} \end{aligned}$$

Thus, rather than using the generated value of T as the estimator from a simulation run, it is better to stop the simulation at time t_0 and use the estimator $t_0 + (\lambda R + X_Q)\frac{E[S]}{1-\lambda E[S]}$. ■

11.6.3 Control Variates

Again suppose we want to use simulation to estimate $E[g(\mathbf{X})]$ where $\mathbf{X} = (X_1, \ldots, X_n)$. But now suppose that for some function f the expected value of $f(\mathbf{X})$ is known—say, $E[f(\mathbf{X})] = \mu$. Then for any constant a we can also use

$$W = g(\mathbf{X}) + a(f(\mathbf{X}) - \mu)$$

as an estimator of $E[g(\mathbf{X})]$. Now,

$$\text{Var}(W) = \text{Var}(g(\mathbf{X})) + a^2 \,\text{Var}(f(\mathbf{X})) + 2a \,\text{Cov}(g(\mathbf{X}), f(\mathbf{X}))$$

Simple calculus shows that the preceding is minimized when

$$a = \frac{-\text{Cov}(f(\mathbf{X}), g(\mathbf{X}))}{\text{Var}(f(\mathbf{X}))}$$

and, for this value of a,

$$\text{Var}(W) = \text{Var}(g(\mathbf{X})) - \frac{[\text{Cov}(f(\mathbf{X}), g(\mathbf{X}))]^2}{\text{Var}(f(\mathbf{X}))}$$

Because $\text{Var}(f(\mathbf{X}))$ and $\text{Cov}(f(\mathbf{X}), g(\mathbf{X}))$ are usually unknown, the simulated data should be used to estimate these quantities.

Dividing the preceding equation by $\text{Var}(g(\mathbf{X}))$ shows that

$$\frac{\text{Var}(W)}{\text{Var}(g(\mathbf{X}))} = 1 - \text{Corr}^2(f(\mathbf{X}), g(\mathbf{X}))$$

where $\text{Corr}(X, Y)$ is the correlation between X and Y. Consequently, the use of a control variate will greatly reduce the variance of the simulation estimator whenever $f(\mathbf{X})$ and $g(\mathbf{X})$ are strongly correlated.

Example 11.20 Consider a continuous-time Markov chain that, upon entering state i, spends an exponential time with rate v_i in that state before making a transition into some other state, with the transition being into state j with probability $P_{i,j}, i \geqslant 0, j \neq i$. Suppose that costs are incurred at rate $C(i) \geqslant 0$ per unit time whenever the chain is in state $i, i \geqslant 0$. With $X(t)$ equal to the state at time t, and α being a constant such that $0 < \alpha < 1$, the quantity

$$W = \int_0^\infty e^{-\alpha t} C(X(t))\, dt$$

represents the total discounted cost. For a given initial state, suppose we want to use simulation to estimate $E[W]$. Whereas at first it might seem that we cannot obtain an unbiased estimator without simulating the continuous-time Markov chain for an infinite amount of time (which is clearly impossible), we can make use of the results of Example 5.1, which gives the equivalent expression for $E[W]$:

$$E[W] = E\left[\int_0^T C(X(t))\,dt\right]$$

where T is an exponential random variable with rate α that is independent of the continuous-time Markov chain. Therefore, we can first generate the value of T, then generate the states of the continuous-time Markov chain up to time T, to obtain the unbiased estimator $\int_0^T C(X(t))\,dt$. Because all the cost rates are nonnegative this estimator is strongly positively correlated with T, which will thus make an effective control variate. ■

Example 11.21 (A Queueing System) Let D_{n+1} denote the delay in queue of the $n+1$ customer in a queueing system in which the interarrival times are independent and identically distributed (i.i.d.) with distribution F having mean μ_F and are independent of the service times, which are i.i.d. with distribution G having mean μ_G. If X_i is the interarrival time between arrival i and $i+1$, and if S_i is the service time of customer i, $i \geqslant 1$, we may write

$$D_{n+1} = g(X_1, \ldots, X_n, S_1, \ldots, S_n)$$

To take into account the possibility that the simulated variables X_i, S_i may by chance be quite different from what might be expected we can let

$$f(X_1, \ldots, X_n, S_1, \ldots, S_n) = \sum_{i=1}^{n} (S_i - X_i)$$

As $E[f(\mathbf{X}, \mathbf{S})] = n(\mu_G - \mu_F)$ we could use

$$g(\mathbf{X}, \mathbf{S}) + a[f(\mathbf{X}, \mathbf{S}) - n(\mu_G - \mu_F)]$$

as an estimator of $E[D_{n+1}]$. Since D_{n+1} and f are both increasing functions of $S_i, -X_i, i = 1, \ldots, n$ it follows from Theorem 11.1 that $f(\mathbf{X}, \mathbf{S})$ and D_{n+1} are positively correlated, and so the simulated estimate of a should turn out to be negative.

If we wanted to estimate the expected sum of the delays in queue of the first $N(T)$ arrivals, then we could use $\sum_{i=1}^{N(T)} S_i$ as our control variable. Indeed as the

arrival process is usually assumed independent of the service times, it follows that

$$E\left[\sum_{i=1}^{N(T)} S_i\right] = E[S]E[N(T)]$$

where $E[N(T)]$ can either be computed by the method suggested in Section 7.8 or estimated from the simulation as in Example 11.18. This control variable could also be used if the arrival process were a nonhomogeneous Poisson with rate $\lambda(t)$; in this case,

$$E[N(T)] = \int_0^T \lambda(t)\, dt$$ ■

11.6.4 Importance Sampling

Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote a vector of random variables having a joint density function (or joint mass function in the discrete case) $f(\mathbf{x}) = f(x_1, \ldots, x_n)$, and suppose that we are interested in estimating

$$\theta = E[h(\mathbf{X})] = \int h(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x}$$

where the preceding is an n-dimensional integral. (If the X_i are discrete, then interpret the integral as an n-fold summation.)

Suppose that a direct simulation of the random vector $\mathbf{X}$, so as to compute values of $h(\mathbf{X})$, is inefficient, possibly because (a) it is difficult to simulate a random vector having density function $f(\mathbf{x})$, or (b) the variance of $h(\mathbf{X})$ is large, or (c) a combination of (a) and (b).

Another way in which we can use simulation to estimate θ is to note that if $g(\mathbf{x})$ is another probability density such that $f(\mathbf{x}) = 0$ whenever $g(\mathbf{x}) = 0$, then we can express θ as

$$\begin{aligned}\theta &= \int \frac{h(\mathbf{x}) f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})\, d\mathbf{x} \\ &= E_g\left[\frac{h(\mathbf{X}) f(\mathbf{X})}{g(\mathbf{X})}\right] \qquad (11.14)\end{aligned}$$

where we have written E_g to emphasize that the random vector $\mathbf{X}$ has joint density $g(\mathbf{x})$.

It follows from Equation (11.14) that θ can be estimated by successively generating values of a random vector $\mathbf{X}$ having density function $g(\mathbf{x})$ and then using as the estimator the average of the values of $h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})$. If a density function

$g(\mathbf{x})$ can be chosen so that the random variable $h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})$ has a small variance then this approach—referred to as *importance sampling*—can result in an efficient estimator of θ.

Let us now try to obtain a feel for why importance sampling can be useful. To begin, note that $f(\mathbf{X})$ and $g(\mathbf{X})$ represent the respective likelihoods of obtaining the vector $\mathbf{X}$ when $\mathbf{X}$ is a random vector with respective densities f and g. Hence, if $\mathbf{X}$ is distributed according to g, then it will usually be the case that $f(\mathbf{X})$ will be small in relation to $g(\mathbf{X})$ and thus when $\mathbf{X}$ is simulated according to g the likelihood ratio $f(\mathbf{X})/g(\mathbf{X})$ will usually be small in comparison to 1. However, it is easy to check that its mean is 1:

$$E_g\left[\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] = \int \frac{f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})\, dx = \int f(\mathbf{x})\, d\mathbf{x} = 1$$

Thus we see that even though $f(\mathbf{X})/g(\mathbf{X})$ is usually smaller than 1, its mean is equal to 1; thus implying that it is occasionally large and so will tend to have a large variance. So how can $h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})$ have a small variance? The answer is that we can sometimes arrange to choose a density g such that those values of $\mathbf{x}$ for which $f(\mathbf{x})/g(\mathbf{x})$ is large are precisely the values for which $h(\mathbf{x})$ is exceedingly small, and thus the ratio $h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})$ is always small. Since this will require that $h(\mathbf{x})$ sometimes be small, importance sampling seems to work best when estimating a small probability; for in this case the function $h(\mathbf{x})$ is equal to 1 when $\mathbf{x}$ lies in some set and is equal to 0 otherwise.

We will now consider how to select an appropriate density g. We will find that the so-called tilted densities are useful. Let $M(t) = E_f[e^{tX}] = \int e^{tx} f(x)\, dx$ be the moment generating function corresponding to a one-dimensional density f.

Definition 11.2 A density function

$$f_t(x) = \frac{e^{tx} f(x)}{M(t)}$$

is called a *tilted* density of f, $-\infty < t < \infty$.

A random variable with density f_t tends to be larger than one with density f when $t > 0$ and tends to be smaller when $t < 0$.

In certain cases the tilted distributions f_t have the same parametric form as does f.

Example 11.22 If f is the exponential density with rate λ then

$$f_t(x) = Ce^{tx}\lambda e^{-\lambda x} = \lambda C e^{-(\lambda - t)x}$$

where $C = 1/M(t)$ does not depend on x. Therefore, for $t \leqslant \lambda$, f_t is an exponential density with rate $\lambda - t$.

If f is a Bernoulli probability mass function with parameter p, then

$$f(x) = p^x(1-p)^{1-x}, \quad x = 0, 1$$

Hence, $M(t) = E_f[e^{tX}] = pe^t + 1 - p$ and so

$$f_t(x) = \frac{1}{M(t)}(pe^t)^x(1-p)^{1-x}$$

$$= \left(\frac{pe^t}{pe^t + 1 - p}\right)^x \left(\frac{1-p}{pe^t + 1 - p}\right)^{1-x} \tag{11.15}$$

That is, f_t is the probability mass function of a Bernoulli random variable with parameter

$$p_t = \frac{pe^t}{pe^t + 1 - p}$$

We leave it as an exercise to show that if f is a normal density with parameters μ and σ^2 then f_t is a normal density with mean $\mu + \sigma^2 t$ and variance σ^2. ■

In certain situations the quantity of interest is the sum of the independent random variables $X_1, \ldots, X_n$. In this case the joint density f is the product of one-dimensional densities. That is,

$$f(x_1, \ldots, x_n) = f_1(x_1) \cdots f_n(x_n)$$

where f_i is the density function of X_i. In this situation it is often useful to generate the X_i according to their tilted densities, with a common choice of t employed.

Example 11.23 Let $X_1, \ldots, X_n$ be independent random variables having respective probability density (or mass) functions f_i, for $i = 1, \ldots, n$. Suppose we are interested in approximating the probability that their sum is at least as large as a, where a is much larger than the mean of the sum. That is, we are interested in

$$\theta = P\{S \geqslant a\}$$

where $S = \sum_{i=1}^n X_i$, and where $a > \sum_{i=1}^n E[X_i]$. Letting $I\{S \geqslant a\}$ equal 1 if $S \geqslant a$ and letting it be 0 otherwise, we have that

$$\theta = E_f[I\{S \geqslant a\}]$$

where $\mathbf{f} = (f_1, \ldots, f_n)$. Suppose now that we simulate X_i according to the tilted mass function $f_{i,t}$, $i = 1, \ldots, n$, with the value of $t, t > 0$ left to be determined. The importance sampling estimator of θ would then be

$$\hat{\theta} = I\{S \geqslant a\} \prod \frac{f_i(X_i)}{f_{i,t}(X_i)}$$

Now,

$$\frac{f_i(X_i)}{f_{i,t}(X_i)} = M_i(t)e^{-tX_i}$$

and so

$$\hat{\theta} = I\{S \geqslant a\}M(t)e^{-tS}$$

where $M(t) = \prod M_i(t)$ is the moment generating function of S. Since $t > 0$ and $I\{S \geqslant a\}$ is equal to 0 when $S < a$, it follows that

$$I\{S \geqslant a\}e^{-tS} \leqslant e^{-ta}$$

and so

$$\hat{\theta} \leqslant M(t)e^{-ta}$$

To make the bound on the estimator as small as possible we thus choose $t, t > 0$, to minimize $M(t)e^{-ta}$. In doing so, we will obtain an estimator whose value on each iteration is between 0 and $\min_t M(t)e^{-ta}$. It can be shown that the minimizing t, call it t^*, is such that

$$E_{t^*}[S] = E_{t^*}\left[\sum_{i=1}^{n} X_i\right] = a$$

where, in the preceding, we mean that the expected value is to be taken under the assumption that the distribution of X_i is f_{i,t^*} for $i = 1, \ldots, n$.

For instance, suppose that $X_1, \ldots, X_n$ are independent Bernoulli random variables having respective parameters p_i, for $i = 1, \ldots, n$. Then, if we generate the X_i according to their tilted mass functions $p_{i,t}, i = 1, \ldots, n$, the importance sampling estimator of $\theta = P\{S \geqslant a\}$ is

$$\hat{\theta} = I\{S \geqslant a\}e^{-tS} \prod_{i=1}^{n} (p_i e^t + 1 - p_i)$$

Since $p_{i,t}$ is the mass function of a Bernoulli random variable with parameter $p_i e^t/(p_i e^t + 1 - p_i)$ it follows that

$$E_t\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \frac{p_i e^t}{p_i e^t + 1 - p_i}$$

The value of t that makes the preceding equal to a can be numerically approximated and then utilized in the simulation.

As an illustration, suppose that $n = 20$, $p_i = 0.4$, and $a = 16$. Then

$$E_t[S] = 20\frac{0.4e^t}{0.4e^t + 0.6}$$

Setting this equal to 16 yields, after a little algebra,

$$e^{t^*} = 6$$

Thus, if we generate the Bernoullis using the parameter

$$\frac{0.4e^{t^*}}{0.4e^{t^*} + 0.6} = 0.8$$

then because

$$M(t^*) = (0.4e^{t^*} + 0.6)^{20} \quad \text{and} \quad e^{-t^*S} = (1/6)^S$$

we see that the importance sampling estimator is

$$\hat{\theta} = I\{S \geqslant 16\}(1/6)^S 3^{20}$$

It follows from the preceding that

$$\hat{\theta} \leqslant (1/6)^{16} 3^{20} = 81/2^{16} = 0.001236$$

That is, on each iteration the value of the estimator is between 0 and 0.001236. Since, in this case, θ is the probability that a binomial random variable with parameters 20, 0.4 is at least 16, it can be explicitly computed with the result $\theta = 0.000317$. Hence, the raw simulation estimator I, which on each iteration takes the value 0 if the sum of the Bernoullis with parameter 0.4 is less than 16 and takes the value 1 otherwise, will have variance

$$\text{Var}(I) = \theta(1 - \theta) = 3.169 \times 10^{-4}$$

On the other hand, it follows from the fact that $0 \leqslant \hat{\theta} \leqslant 0.001236$ that (see Exercise 33)

$$\text{Var}(\hat{\theta}) \leqslant 2.9131 \times 10^{-7}$$ ■

Example 11.24 Consider a single-server queue in which the times between successive customer arrivals have density function f and the service times have density g. Let D_n denote the amount of time that the nth arrival spends waiting in queue and suppose we are interested in estimating $\alpha = P\{D_n \geqslant a\}$ when a is much larger than $E[D_n]$. Rather than generating the successive interarrival and service times according to f and g, respectively, they should be generated according to the densities f_{-t} and g_t, where t is a positive number to be determined. Note that using these distributions as opposed to f and g will result in smaller interarrival times (since $-t < 0$) and larger service times. Hence, there will be a greater chance that $D_n > a$ than if we had simulated using the densities f and g. The importance sampling estimator of α would then be

$$\hat{\alpha} = I\{D_n > a\}e^{t(S_n - Y_n)}[M_f(-t)M_g(t)]^n$$

where S_n is the sum of the first n interarrival times, Y_n is the sum of the first n service times, and M_f and M_g are the moment generating functions of the densities f and g, respectively. The value of t used should be determined by experimenting with a variety of different choices. ■

11.7 Determining the Number of Runs

Suppose that we are going to use simulation to generate r independent and identically distributed random variables $Y^{(1)}, \ldots, Y^{(r)}$ having mean μ and variance σ^2. We are then going to use

$$\bar{Y}_r = \frac{Y^{(1)} + \cdots + Y^{(r)}}{r}$$

as an estimate of μ. The precision of this estimate can be measured by its variance

$$\begin{aligned}\text{Var}(\bar{Y}_r) &= E[(\bar{Y}_r - \mu)^2] \\ &= \sigma^2/r\end{aligned}$$

Hence, we would want to choose r, the number of necessary runs, large enough so that σ^2/r is acceptably small. However, the difficulty is that σ^2 is not known in

advance. To get around this, you should initially simulate k runs (where $k \geqslant 30$) and then use the simulated values $Y^{(1)}, \ldots, Y^{(k)}$ to estimate σ^2 by the sample variance

$$\sum_{i=1}^{k} \left(Y^{(i)} - \bar{Y}_k\right)^2 / (k-1)$$

Based on this estimate of σ^2 the value of r that attains the desired level of precision can now be determined and an additional $r - k$ runs can be generated.

11.8 Generating from the Stationary Distribution of a Markov Chain

11.8.1 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. Whereas we could *approximately* generate such a random variable by arbitrarily choosing an initial state, simulating the resulting Markov chain for a large fixed number of time periods, and then choosing the final state as the value of the random variable, we will now 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 backward in time in this manner, we will at some time, 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 11.25 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. ■

Remark 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*.

11.8.2 Another Approach

Consider a Markov chain whose state space is the nonnegative integers. Suppose the chain has stationary probabilities, and denote them by $\pi_i, i \geqslant 0$. We now present another way of simulating a random variable whose distribution is given by the $\pi_i, i \geqslant 0$, which can be utilized if the chain satisfies the following property. Namely, that for some state, which we will call state 0, and some positive number α

$$P_{i,0} \geqslant \alpha > 0$$

for all states i. That is, whatever the current state, the probability that the next state will be 0 is at least some positive value α.

To simulate a random variable distributed according to the stationary probabilities, start by simulating the Markov chain in the obvious manner. Namely, whenever the chain is in state i, generate a random variable that is equal to j with probability $P_{i,j}, j \geqslant 0$, and then set the next state equal to the generated value of this random variable. In addition, however, whenever a transition into state 0 occurs a coin, whose probability of coming up heads depends on the state from which the transition occurred, is flipped. Specifically, if the transition into state 0 was from state i, then the coin flipped has probability $\alpha/P_{i,0}$ of coming up heads. Call such a coin an i-coin, $i \geqslant 0$. If the coin comes up heads then we say that an event has occurred. Consequently, each transition of the Markov chain results in an event with probability α, implying that events occur at rate α. Now say that an event is an i-event if it resulted from a transition out of state i; that is, an event is an i-event if it resulted from the flip of an i-coin. Because π_i is the proportion

of transitions that are out of state i, and each such transition will result in an i-event with probability α, it follows that the rate at which i-events occur is $\alpha\pi_i$. Therefore, the proportion of all events that are i-events is $\alpha\pi_i/\alpha = \pi_i, i \geqslant 0$.

Now, suppose that $X_0 = 0$. Fix i, and let I_j equal 1 if the j^{th} event that occurs is an i-event, and let I_j equal 0 otherwise. Because an event always leaves the chain in state 0 it follows that $I_j, j \geqslant 1$, are independent and identically distributed random variables. Because the proportion of the I_j that are equal to 1 is π_i, we see that

$$\begin{aligned}\pi_i &= \lim_{n\to\infty} \frac{I_1 + \cdots + I_n}{n} \\ &= E[I_1] \\ &= P(I_1 = 1)\end{aligned}$$

where the second equality follows from the strong law of large numbers. Hence, if we let

$$T = \min\{n > 0 : \text{an event occurs at time } n\}$$

denote the time of the first event, then it follows from the preceding that

$$\pi_i = P(I_1 = 1) = P(X_{T-1} = i)$$

As the preceding is true for all states i, it follows that X_{T-1}, the state of the Markov chain at time $T - 1$, has the stationary distribution.

Exercises

*1. Suppose it is relatively easy to simulate from the distributions F_i, $i = 1, 2, \ldots, n$. If n is small, how can we simulate from

$$F(x) = \sum_{i=1}^{n} P_i F_i(x), \quad P_i \geqslant 0, \qquad \sum_i P_i = 1?$$

Give a method for simulating from

$$F(x) = \begin{cases} \dfrac{1 - e^{-2x} + 2x}{3}, & 0 < x < 1 \\ \dfrac{3 - e^{-2x}}{3}, & 1 < x < \infty \end{cases}$$

2. Give a method for simulating a negative binomial random variable.

*3. Give a method for simulating a hypergeometric random variable.

4. Suppose we want to simulate a point located at random in a circle of radius r centered at the origin. That is, we want to simulate X, Y having joint density

$$f(x, y) = \frac{1}{\pi r^2}, \quad x^2 + y^2 \leqslant r^2$$

   (a) Let $R = \sqrt{X^2 + Y^2}$, $\theta = \tan^{-1} Y/X$ denote the polar coordinates. Compute the joint density of R, θ and use this to give a simulation method. Another method for simulating X, Y is as follows:

   Step 1: Generate independent random numbers U_1, U_2 and set $Z_1 = 2rU_1 - r, Z_2 = 2rU_2 - r$. Then Z_1, Z_2 is uniform in the square whose sides are of length $2r$ and which encloses, the circle of radius r (see Figure 11.6).

   Step 2: If (Z_1, Z_2) lies in the circle of radius r—that is, if $Z_1^2 + Z_2^2 \leqslant r^2$—set $(X, Y) = (Z_1, Z_2)$. Otherwise return to step 1.

   (b) Prove that this method works, and compute the distribution of the number of random numbers it requires.

5. Suppose it is relatively easy to simulate from F_i for each $i = 1, \ldots, n$. How can we simulate from
   (a) $F(x) = \prod_{i=1}^{n} F_i(x)$?
   (b) $F(x) = 1 - \prod_{i=1}^{n}(1 - F_i(x))$?
   (c) Give two methods for simulating from the distribution $F(x) = x^n$, $0 < x < 1$.

*6. In Example 11.4 we simulated the absolute value of a standard normal by using the Von Neumann rejection procedure on exponential random variables with rate 1. This raises the question of whether we could obtain a more efficient algorithm by using a different exponential density—that is, we could use the density $g(x) = \lambda e^{-\lambda x}$. Show that the mean number of iterations needed in the rejection scheme is minimized when $\lambda = 1$.

7. Give an algorithm for simulating a random variable having density function

$$f(x) = 30(x^2 - 2x^3 + x^4), \quad 0 < x < 1$$

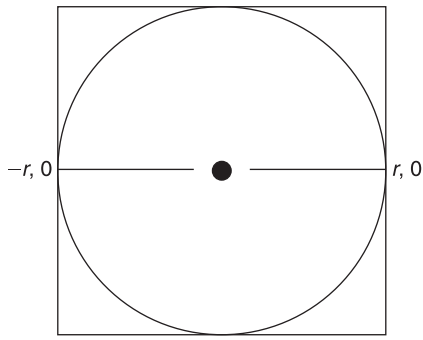

Figure 11.6

8. Consider the technique of simulating a gamma (n, λ) random variable by using the rejection method with g being an exponential density with rate λ/n.
   (a) Show that the average number of iterations of the algorithm needed to generate a gamma is $n^n e^{1-n}/(n-1)!$.
   (b) Use Stirling's approximation to show that for large n the answer to part (a) is approximately equal to $e[(n-1)/(2\pi)]^{1/2}$.
   (c) Show that the procedure is equivalent to the following:

   Step 1: Generate Y_1 and Y_2, independent exponentials with rate 1.

   Step 2: If $Y_1 < (n-1)[Y_2 - \log(Y_2) - 1]$, return to step 1.

   Step 3: Set $X = nY_2/\lambda$.

   (d) Explain how to obtain an independent exponential along with a gamma from the preceding algorithm.
9. Set up the alias method for simulating from a binomial random variable with parameters $n = 6$, $p = 0.4$.
10. Explain how we can number the $\mathbf{Q}^{(k)}$ in the alias method so that k is one of the two points that $\mathbf{Q}^{(k)}$ gives weight.

    Hint: Rather than giving the initial $\mathbf{Q}$ the name $\mathbf{Q}^{(1)}$, what else could we call it?
11. Complete the details of Example 11.10.
12. Let $X_1, \ldots, X_k$ be independent with

$$P\{X_i = j\} = \frac{1}{n}, \quad j = 1, \ldots, n, \ i = 1, \ldots, k$$

    If D is the number of distinct values among $X_1, \ldots, X_k$ show that

$$E[D] = n\left[1 - \left(\frac{n-1}{n}\right)^k\right]$$

$$\approx k - \frac{k^2}{2n} \quad \text{when } \frac{k^2}{n} \text{ is small}$$

13. *The Discrete Rejection Method:* Suppose we want to simulate X having probability mass function $P\{X = i\} = P_i, i = 1, \ldots, n$ and suppose we can easily simulate from the probability mass function $Q_i, \sum_i Q_i = 1, Q_i \geqslant 0$. Let C be such that $P_i \leqslant CQ_i, i = 1, \ldots, n$. Show that the following algorithm generates the desired random variable:

    Step 1: Generate Y having mass function Q and U an independent random number.

    Step 2: If $U \leqslant P_Y/CQ_Y$, set $X = Y$. Otherwise return to step 1.
14. *The Discrete Hazard Rate Method:* Let X denote a nonnegative integer valued random variable. The function $\lambda(n) = P\{X = n \mid X \geqslant n\}$, $n \geqslant 0$, is called the *discrete hazard rate function.*
    (a) Show that $P\{X = n\} = \lambda(n) \prod_{i=0}^{n-1} (1 - \lambda(i))$.
    (b) Show that we can simulate X by generating random numbers $U_1, U_2, \ldots$ stopping at

$$X = \min\{n: U_n \leqslant \lambda(n)\}$$

(c) Apply this method to simulating a geometric random variable. Explain, intuitively, why it works.

(d) Suppose that $\lambda(n) \leqslant p < 1$ for all n. Consider the following algorithm for simulating X and explain why it works: Simulate $X_i, U_i, i \geqslant 1$ where X_i is geometric with mean $1/p$ and U_i is a random number. Set $S_k = X_1 + \cdots + X_k$ and let

$$X = \min\{S_k: U_k \leqslant \lambda(S_k)/p\}$$

15. Suppose you have just simulated a normal random variable X with mean μ and variance σ^2. Give an easy way to generate a second normal variable with the same mean and variance that is negatively correlated with X.

*16. Suppose n balls having weights $w_1, w_2, \ldots, w_n$ are in an urn. These balls are sequentially removed in the following manner: At each selection, a given ball in the urn is chosen with a probability equal to its weight divided by the sum of the weights of the other balls that are still in the urn. Let $I_1, I_2, \ldots, I_n$ denote the order in which the balls are removed—thus $I_1, \ldots, I_n$ is a random permutation with weights.

(a) Give a method for simulating $I_1, \ldots, I_n$.

(b) Let X_i be independent exponentials with rates $w_i, i = 1, \ldots, n$. Explain how X_i can be utilized to simulate $I_1, \ldots, I_n$.

17. *Order Statistics:* Let $X_1, \ldots, X_n$ be i.i.d. from a continuous distribution F, and let $X_{(i)}$ denote the ith smallest of $X_1, \ldots, X_n, i = 1, \ldots, n$. Suppose we want to simulate $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$. One approach is to simulate n values from F, and then order these values. However, this ordering, or *sorting*, can be time consuming when n is large.

(a) Suppose that $\lambda(t)$, the hazard rate function of F, is bounded. Show how the hazard rate method can be applied to generate the n variables in such a manner that no sorting is necessary.

Suppose now that F^{-1} is easily computed.

(b) Argue that $X_{(1)}, \ldots, X_{(n)}$ can be generated by simulating $U_{(1)} < U_{(2)} < \cdots < U_{(n)}$—the ordered values of n independent random numbers—and then setting $X_{(i)} = F^{-1}(U_{(i)})$. Explain why this means that $X_{(i)}$ can be generated from $F^{-1}(\beta_i)$ where β_i is beta with parameters $i, n + i + 1$.

(c) Argue that $U_{(1)}, \ldots, U_{(n)}$ can be generated, without any need for sorting, by simulating i.i.d. exponentials $Y_1, \ldots, Y_{n+1}$ and then setting

$$U_{(i)} = \frac{Y_1 + \cdots + Y_i}{Y_1 + \cdots + Y_{n+1}}, \quad i = 1, \ldots, n$$

Hint: Given the time of the $(n + 1)$st event of a Poisson process, what can be said about the set of times of the first n events?

(d) Show that if $U_{(n)} = y$ then $U_{(1)}, \ldots, U_{(n-1)}$ has the same joint distribution as the order statistics of a set of $n - 1$ uniform $(0, y)$ random variables.

(e) Use part (d) to show that $U_{(1)}, \ldots, U_{(n)}$ can be generated as follows:
Step 1: Generate random numbers $U_1, \ldots, U_n$.

Step 2: Set

$$U_{(n)} = U_1^{1/n}, \qquad U_{(n-1)} = U_{(n)}(U_2)^{1/(n-1)},$$
$$U_{(j-1)} = U_{(j)}(U_{n-j+2})^{1/(j-1)}, \quad j = 2, \ldots, n-1$$

18. Let $X_1, \ldots, X_n$ be independent exponential random variables each having rate 1. Set

$$W_1 = X_1/n,$$
$$W_i = W_{i-1} + \frac{X_i}{n-i+1}, \quad i = 2, \ldots, n$$

Explain why $W_1, \ldots, W_n$ has the same joint distribution as the order statistics of a sample of n exponentials each having rate 1.

19. Suppose we want to simulate a large number n of independent exponentials with rate 1—call them $X_1, X_2, \ldots, X_n$. If we were to employ the inverse transform technique we would require one logarithmic computation for each exponential generated. One way to avoid this is to first simulate S_n, a gamma random variable with parameters $(n, 1)$ (say, by the method of Section 11.3.3). Now interpret S_n as the time of the nth event of a Poisson process with rate 1 and use the result that given S_n the set of the first $n-1$ event times is distributed as the set of $n-1$ independent uniform $(0, S_n)$ random variables. Based on this, explain why the following algorithm simulates n independent exponentials:

Step 1: Generate S_n, a gamma random variable with parameters $(n, 1)$.
Step 2: Generate $n-1$ random numbers $U_1, U_2, \ldots, U_{n-1}$.
Step 3: Order the $U_i, i = 1, \ldots, n-1$ to obtain $U_{(1)} < U_{(2)} < \cdots < U_{(n-1)}$.
Step 4: Let $U_{(0)} = 0, U_{(n)} = 1$, and set $X_i = S_n(U_{(i)} - U_{(i-1)}), i = 1, \ldots, n$.

When the ordering (step 3) is performed according to the algorithm described in Section 11.5, the preceding is an efficient method for simulating n exponentials when all n are simultaneously required. If memory space is limited, however, and the exponentials can be employed sequentially, discarding each exponential from memory once it has been used, then the preceding may not be appropriate.

20. Consider the following procedure for randomly choosing a subset of size k from the numbers $1, 2, \ldots, n$: Fix p and generate the first n time units of a renewal process whose interarrival distribution is geometric with mean $1/p$—that is, $P\{\text{interarrival time} = k\} = p(1-p)^{k-1}, k = 1, 2, \ldots$. Suppose events occur at times $i_1 < i_2 < \cdots < i_m \leqslant n$. If $m = k$, stop; $i_1, \ldots, i_m$ is the desired set. If $m > k$, then randomly choose (by some method) a subset of size k from $i_1, \ldots, i_m$ and then stop. If $m < k$, take $i_1, \ldots, i_m$ as part of the subset of size k and then select (by some method) a random subset of size $k-m$ from the set $\{1, 2, \ldots, n\} - \{i_1, \ldots, i_m\}$. Explain why this algorithm works. As $E[N(n)] = np$ a reasonable choice of p is to take $p \approx k/n$. (This approach is due to Dieter.)

21. Consider the following algorithm for generating a random permutation of the elements $1, 2, \ldots, n$. In this algorithm, $P(i)$ can be interpreted as the element in position i.

Step 1: Set $k = 1$.

Step 2: Set $P(1) = 1$.

Step 3: If $k = n$, stop. Otherwise, let $k = k + 1$.

Step 4: Generate a random number U, and let

$$P(k) = P([kU] + 1),$$
$$P([kU] + 1) = k.$$

Go to step 3.

(a) Explain in words what the algorithm is doing.
(b) Show that at iteration k—that is, when the value of $P(k)$ is initially set—that $P(1), P(2), \ldots, P(k)$ is a random permutation of $1, 2, \ldots, k$.

Hint: Use induction and argue that

$$\begin{aligned} &P_k\{i_1, i_2, \ldots, i_{j-1}, k, i_j, \ldots, i_{k-2}, i\} \\ &\quad = P_{k-1}\{i_1, i_2, \ldots, i_{j-1}, i, i_j, \ldots, i_{k-2}\}\frac{1}{k} \\ &\quad = \frac{1}{k!} \quad \text{by the induction hypothesis} \end{aligned}$$

The preceding algorithm can be used even if n is not initially known.

22. Verify that if we use the hazard rate approach to simulate the event times of a nonhomogeneous Poisson process whose intensity function $\lambda(t)$ is such that $\lambda(t) \leqslant \lambda$, then we end up with the approach given in method 1 of Section 11.5.

*23. For a nonhomogeneous Poisson process with intensity function $\lambda(t)$, $t \geqslant 0$, where $\int_0^\infty \lambda(t)\,dt = \infty$, let $X_1, X_2, \ldots$ denote the sequence of times at which events occur.
(a) Show that $\int_0^{X_1} \lambda(t)\,dt$ is exponential with rate 1.
(b) Show that $\int_{X_{i-1}}^{X_i} \lambda(t)\,dt, i \geqslant 1$, are independent exponentials with rate 1, where $X_0 = 0$.

In words, independent of the past, the additional amount of hazard that must be experienced until an event occurs is exponential with rate 1.

24. Give an efficient method for simulating a nonhomogeneous Poisson process with intensity function

$$\lambda(t) = b + \frac{1}{t + a}, \quad t \geqslant 0$$

25. Let (X, Y) be uniformly distributed in a circle of radius r about the origin. That is, their joint density is given by

$$f(x, y) = \frac{1}{\pi r^2}, \quad 0 \leqslant x^2 + y^2 \leqslant r^2$$

Let $R = \sqrt{X^2 + Y^2}$ and $\theta = \arctan Y/X$ denote their polar coordinates. Show that R and θ are independent with θ being uniform on $(0, 2\pi)$ and $P\{R < a\} = a^2/r^2$, $0 < a < r$.

26. Let R denote a region in the two-dimensional plane. Show that for a two-dimensional Poisson process, given that there are n points located in R, the points are independently and uniformly distributed in R—that is, their density is $f(x, y) = c, (x, y) \in R$ where c is the inverse of the area of R.
27. Let $X_1, \ldots, X_n$ be independent random variables with $E[X_i] = \theta$, $\text{Var}(X_i) = \sigma_i^2$ $i = 1, \ldots, n$, and consider estimates of θ of the form $\sum_{i=1}^n \lambda_i X_i$ where $\sum_{i=1}^n \lambda_i = 1$. Show that $\text{Var}\left(\sum_{i=1}^n \lambda_i X_i\right)$ is minimized when

$$\lambda_i = (1/\sigma_i^2) \Big/ \left(\sum_{j=1}^n 1/\sigma_j^2\right), \quad i = 1, \ldots, n.$$

Possible Hint: If you cannot do this for general n, try it first when $n = 2$.

The following two problems are concerned with the estimation of $\int_0^1 g(x)\,dx = E[g(U)]$ where U is uniform $(0, 1)$.

28. *The Hit–Miss Method:* Suppose g is bounded in $[0, 1]$—for instance, suppose $0 \leqslant g(x) \leqslant b$ for $x \in [0, 1]$. Let U_1, U_2 be independent random numbers and set $X = U_1, Y = bU_2$—so the point (X, Y) is uniformly distributed in a rectangle of length 1 and height b. Now set

$$I = \begin{cases} 1, & \text{if } Y < g(X) \\ 0, & \text{otherwise} \end{cases}$$

That is, accept (X, Y) if it falls in the shaded area of Figure 11.7.

(a) Show that $E[bI] = \int_0^1 g(x)\,dx$.
(b) Show that $\text{Var}(bI) \geqslant \text{Var}(g(U))$, and so hit–miss has larger variance than simply computing g of a random number.

29. *Stratified Sampling:* Let $U_1, \ldots, U_n$ be independent random numbers and set $\bar{U}_i = (U_i + i - 1)/n$, $i = 1, \ldots, n$. Hence, $\bar{U}_i, i \geqslant 1$, is uniform on $((i-1)/n, i/n)$. $\sum_{i=1}^n g(\bar{U}_i)/n$ is called the stratified sampling estimator of $\int_0^1 g(x)\,dx$.

(a) Show that $E[\sum_{i=1}^n g(\bar{U}_i)/n] = \int_0^1 g(x)\,dx$.
(b) Show that $\text{Var}[\sum_{i=1}^n g(\bar{U}_i)/n] \leqslant \text{Var}[\sum_{i=1}^n g(U_i)/n]$.

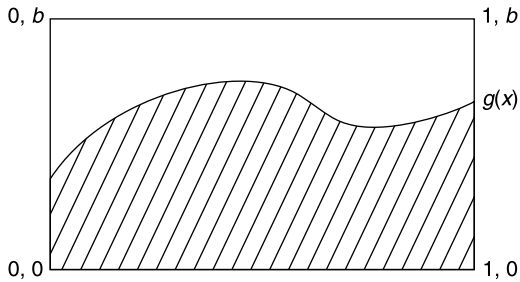

Figure 11.7

Hint: Let U be uniform $(0,1)$ and define N by $N = i$ if $(i-1)/n < U < i/n$, $i = 1, \ldots, n$. Now use the conditional variance formula to obtain

$$\begin{aligned}\text{Var}(g(U)) &= E[\text{Var}(g(U)|N)] + \text{Var}(E[g(U)|N]) \\ &\geqslant E[\text{Var}(g(U)|N)] \\ &= \sum_{i=1}^{n} \frac{\text{Var}(g(U)|N=i)}{n} = \sum_{i=1}^{n} \frac{\text{Var}[g(\bar{U}_i)]}{n}\end{aligned}$$

30. If f is the density function of a normal random variable with mean μ and variance σ^2, show that the tilted density f_t is the density of a normal random variable with mean $\mu + \sigma^2 t$ and variance σ^2.

31. Consider a queueing system in which each service time, independent of the past, has mean μ. Let W_n and D_n denote, respectively, the amounts of time customer n spends in the system and in queue. Hence, $D_n = W_n - S_n$ where S_n is the service time of customer n. Therefore,

$$E[D_n] = E[W_n] - \mu$$

If we use simulation to estimate $E[D_n]$, should we
(a) use the simulated data to determine D_n, which is then used as an estimate of $E[D_n]$; or
(b) use the simulated data to determine W_n and then use this quantity minus μ as an estimate of $E[D_n]$?
Repeat for when we want to estimate $E[W_n]$.

*32. Show that if X and Y have the same distribution then

$$\text{Var}((X+Y)/2) \leqslant \text{Var}(X)$$

Hence, conclude that the use of antithetic variables can never increase variance (though it need not be as efficient as generating an independent set of random numbers).

33. If $0 \leqslant X \leqslant a$, show that
(a) $E[X^2] \leqslant aE[X]$,
(b) $\text{Var}(X) \leqslant E[X](a - E[X])$,
(c) $\text{Var}(X) \leqslant a^2/4$.

34. Suppose in Example 11.19 that no new customers are allowed in the system after time t_0. Give an efficient simulation estimator of the expected additional time after t_0 until the system becomes empty.

35. Suppose we are able to simulate independent random variables X and Y. If we simulate $2k$ independent random variables $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$, where the X_i have the same distribution as does X, and the Y_j have the same distribution as does Y, how would you use them to estimate $P(X < Y)$?

36. If U_1, U_2, U_3 are independent uniform $(0,1)$ random variables, find $P\left(\prod_{i=1}^{3} U_i > 0.1\right)$.

Hint: Relate the desired probability to one about a Poisson process.

References

[1] J. Banks and J. Carson, "Discrete Event System Simulation," Prentice Hall, Englewood Cliffs, New Jersey, 1984.

[2] G. Fishman, "Principles of Discrete Event Simulation," John Wiley, New York, 1978.

[3] D. Knuth, "Semi Numerical Algorithms," Vol. 2 of *The Art of Computer Programming*, Second Edition, Addison-Wesley, Reading, Massachusetts, 1981.

[4] A. Law and W. Kelton, "Simulation Modelling and Analysis," Second Edition, McGraw-Hill, New York, 1992.

[5] J. Propp and D. Wilson, "Coupling From The Past: A User's Guide," Workshop on Microsurveys in Discrete Probability, Princeton, NJ, 1997.

[6] S. Ross, "Simulation," Fourth Edition, Academic Press, San Diego, 2006.

[7] R. Rubenstein, "Simulation and the Monte Carlo Method," John Wiley, New York, 1981.