

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 $X_1, X_2, \dots$ having distribution F , stopping at

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

The $X_i, i \geq 1$, 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 $X_1, X_1 + X_2, \dots, X_1 + \dots + X_{N-1}$.

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 $U_1, \dots, U_n$ —and $\{tU_1, \dots, tU_n\}$ 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 $s < t$,

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

and so to compute the function $N(s), s \leq t$, it is best to first order the values $U_i, i = 1, \dots, n$ 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 $\lambda(t)$, $0 \leq t < \infty$.

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

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

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 $\lambda(t)$, $0 \leq t \leq T$. 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 $X_1, U_1, X_2, U_2, \dots$ 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 $j = 1, \dots, N - 1$,

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

and set

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

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.

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

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 , $0 < t_1 < t_2 < \dots < t_k < T$, $\lambda_1, \dots, \lambda_{k+1}$, such that

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

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 $t_{i-1} \leq t < t_i$).

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 $U \leq \lambda(t)/\lambda_I$. Return to step 2.

Step 4: (Step reached if $t + X \geq t_I$). 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

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

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.

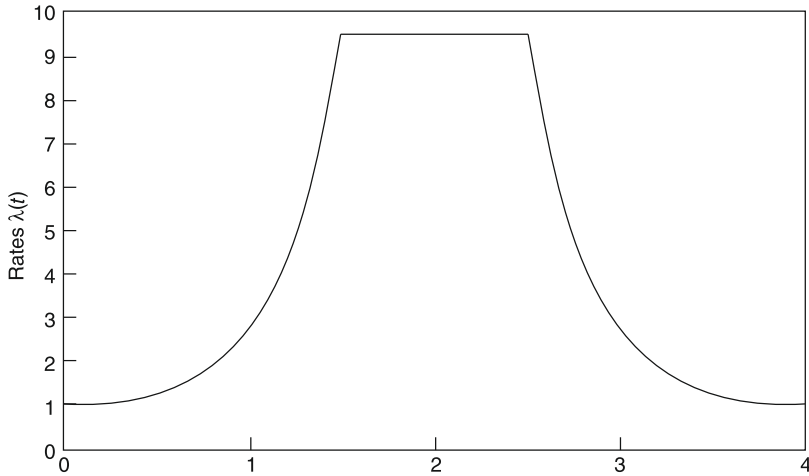


Figure 11.3

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 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

$$\begin{aligned} \lambda_1(t) &\equiv 1, \\ \lambda_2(t) &= \begin{cases} \lambda(t) - 1, & 0 < t < 1 \\ e - 1, & 1 < t < 3 \\ \lambda(t) - 1, & 3 < t < 4 \end{cases} \\ \lambda_3(t) &= \begin{cases} \lambda(t) - e, & 1 < t < 1.5 \\ e^{2.25} - e, & 1.5 < t < 2.5 \\ \lambda(t) - e, & 2.5 < t < 3 \\ 0, & 3 < t < 4 \end{cases} \end{aligned}$$

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

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

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

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

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

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

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

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

or, equivalently, if

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

Method 3. Simulating the Event Times

The third method we shall present for simulating a nonhomogeneous Poisson process having intensity function $\lambda(t)$, $t \geq 0$ is probably the most basic approach—namely, to simulate the successive event times. So let $X_1, X_2, \dots$ 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 $X_1, \dots, X_{i-1}$.

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

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

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 $X_1, X_2, \dots$ 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

$\lambda(t) \leq \lambda$ 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 $\lambda(x) = 1/(x + a)$, $x \geq 0$. 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 $X_1, X_2, \dots$ by generating $U_1, U_2, \dots$ 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,

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

■

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 $\mathbf{O}$ 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 $\mathbf{O}$. Let R_i , $i \geq 1$, denote the distance between $\mathbf{O}$ and its i th nearest Poisson point, and let $C(a)$ denote the circle of radius a centered at $\mathbf{O}$. Then

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

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

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

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

Proposition 11.6 With $R_0 = 0$,

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

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 $\mathbf{O}$:

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

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

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

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

Step 3: Generate independent uniform $(0, 1)$ random variables $U_1, \dots, U_{N-1}$.

Step 4: Return the $N - 1$ Poisson points in $C(r)$ whose polar coordinates are

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

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

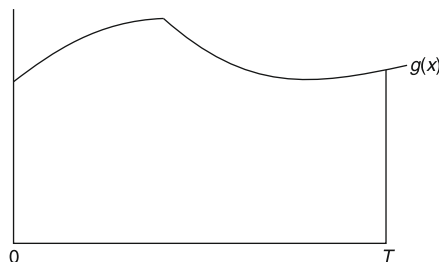


Figure 11.4

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 $\mathbf{O}$ 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 $X_1 < X_2 < \dots$ denote the successive projections of the Poisson points on the x -axis, then analogous to Proposition 11.6, it will follow that (with $X_0 = 0$) $\lambda \int_{X_{i-1}}^{X_i} g(x) dx, i \geq 1$, will be independent exponentials with rate 1. Hence, we should simulate $\epsilon_1, \epsilon_2, \dots$, independent exponentials with rate 1, stopping at

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

and determine $X_1, \dots, X_{N-1}$ 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 $U_1, \dots, U_{N-1}$ —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 $(X_i, U_i g(X_i)), i = 1, \dots, N - 1$.