Additional Variance Reduction Techniques

Introduction

In this chapter we give some additional variation reduction techniques that are not as standard as the ones in the previous chapter. Section 10.1 presents the conditional Bernoulli sampling method which, when applicable, can be used to estimate p, the probability that at least one of a specified set of events occurs. The method is particularly powerful when p is small. Section 10.2 presents the normalized importance sampling technique, which extends the importance sampling idea to situations where the distribution of the random vector to be generated is not completely specified. Section 10.3 introduces Latin Hypercube sampling, a variance reduction technique inspired by the idea of stratified sampling.

10.1 The Conditional Bernoulli Sampling Method

The conditional Bernoulli sampling method (CBSM) is a powerful approach that can often be used when estimating the probability of a union of events. That is, suppose for given events $A_1, \ldots, A_m$ we are interested in using simulation to estimate

$$p = P(\cup_{i=1}^{m} A_i) = P(\sum_{i=1}^{m} X_i > 0),$$

where

$$X_i = \begin{cases} 1, & \text{if } A_i \text{ occurs} \\ 0, & \text{if otherwise.} \end{cases}$$

Let $\lambda_i = P(X_i = 1) = P(A_i)$. Assuming that λ_i is known for all $i = 1, \ldots, m$ and that we are able to generate the values of $X_1, \ldots, X_m$ conditional on a specified

Simulation. DOI: http://dx.doi.org/10.1016/B978-0-12-415825-2.00010-3
© 2013 Elsevier Inc. All rights reserved.

one of them being equal to 1, the CBSM will yield an unbiased estimator of p that will have a small variance when $\sum_{i=1}^{m} P(A_i)$ is small.

Before presenting the method we need some preliminary material. To begin, let $S = \sum_{i=1}^{m} X_i$, and set $\lambda = E[S] = \sum_{i=1}^{m} \lambda_i$. Let R be an arbitrary random variable, and suppose that I is independent of $R, X_1, \ldots, X_m$ and is such that

$$P(I = i) = 1/m, \quad i = 1, \ldots, m$$

That is, I is a discrete uniform random variable on $1, \ldots, m$ that is independent of the other random variables.

The following identity is the key to the results of this section.

Proposition

(a) $P\{I = i | X_I = 1\} = \lambda_i/\lambda$
(b) $E[SR] = \lambda E[R | X_I = 1]$
(c) $P\{S > 0\} = \lambda E\left[\frac{1}{S} | X_I = 1\right]$

Proof To prove (a), note that

$$P\{I = i | X_I = 1\} = \frac{P\{X_I = 1 | I = i\} P\{I = i\}}{\sum_i P\{X_I = 1 | I = i\} P\{I = i\}}$$

Now,

$$\begin{aligned} P\{X_I = 1 | I = i\} &= P\{X_i = 1 | I = i\} \\ &= P\{X_i = 1\} \quad \text{by independence} \\ &= \lambda_i \end{aligned}$$

which completes the proof of (a). To prove (b), reason as follows:

$$\begin{aligned} E[SR] &= E\left[R \sum_i X_i\right] \\ &= \sum_i E[RX_i] \\ &= \sum_i \{E[RX_i | X_i = 1]\lambda_i + E[RX_i | X_i = 0](1 - \lambda_i)\} \\ &= \sum_i \lambda_i E[R | X_i = 1] \end{aligned} \tag{10.1}$$

Also,

$$
\begin{aligned}
E[R|X_I = 1] &= \sum_i E[R|X_I = 1, I = i]P\{I = i|X_I = 1\} \\
&= \sum_i E[R|X_i = 1, I = i]\lambda_i/\lambda \quad \text{by (a)} \\
&= \sum_i E[R|X_i = 1]\lambda_i/\lambda \qquad (10.2)
\end{aligned}
$$

Combining Equations (10.1) and (10.2) proves (b).

To prove (c), define R to equal 0 if $S = 0$ and to equal $1/S$ if $S > 0$. Then,

$$
E[SR] = P\{S > 0\} \quad \text{and} \quad E[R|X_I = 1] = E\left[\frac{1}{S}\middle| X_I = 1\right]
$$

and so (c) follows directly from (b).

Using the preceding proposition we are now in position to give the CBSM.

The Conditional Bernoulli Sampling Method for estimating $p = P(S > 0)$

$$
\lambda_i = P(X_i = 1) = 1 - P(X_i = 0), \quad \lambda = \sum_{i=1}^{m} \lambda_i, \quad S = \sum_{i=1}^{m} X_i.
$$

1. Generate J such that $P(J = i) = \lambda_i/\lambda, i = 1, \ldots, m$. Suppose the generated value of J is j.
2. Set $X_j = 1$
3. Generate the vector $X_i, i \neq j$, conditional on $X_j = 1$
4. Let $S^* = \sum_{i=1}^{m} X_i$ and return the unbiased estimator λ/S^*.

Note that because $S^* \geq 1$ it follows that

$$
0 \leqslant \lambda/S^* \leqslant \lambda
$$

indicating (see Exercise **35** of Chapter 9) that

$$
\text{Var}(\lambda/S^*) \leqslant \lambda^2/4
$$

Remarks

(a) With I being equally likely to be any of the values $1, \ldots, m$, note that J is distributed as I conditional on the event that $X_I = 1$.

(b) As noted in the preceding, the CBSM estimator is always less than or equal to λ, which is the Boole inequality bound on the probability of a union. Provided that $\lambda \leqslant 1$ the CBSM estimator will have a smaller variance than the raw simulation estimator.

(c) Typically, if p is small then λ is of the order of p. Consequently, the variance of the CBSM estimator is typically of the order p^2, whereas the variance of the raw simulation estimator is $p(1-p) \approx p$.

(d) When m is not too large, the CBSM can be improved by using stratified sampling. That is, if you are planning to do r simulation runs then there is no need to generate the value of J. Either use proportional sampling and do $r\lambda_j/\lambda$ runs using $J = j$ for each $j = 1, \ldots, m$, or, when practicable, do a small simulation study to estimate the quantities $\sigma_j^2 = \mathrm{Var}(1/S^*|J = j)$, and then perform $r\frac{\lambda_j \sigma_j}{\sum_{i=1}^m \lambda_i \sigma_i}$ runs using $J = j, j = 1, \ldots, m$. Also, if the simulation experiment is performed without any stratification then post-stratiication should be considered. □

We will now use the CBSM to estimate (a) the probability that more than k coupons are needed in the classical coupon collectors problem, (b) the failure probability of a system, and (c) the probability that a specified pattern appears within a specified time frame.

Example 10a The Coupon Collecting Problem Suppose there are m types of coupons, and that each new coupon collected is type i with probability $p_i, \sum_{i=1}^m p_i = 1$. Let T be the number of coupons until our collection contains at least one of each type, and suppose we are interested in estimating $p = P(T > k)$ when this probability is small. Let $N_i, i = 1, \ldots, m$ denote the number of type i coupons among the first k collected and let $X_i = I\{N_i = 0\}$ be the indicator of the event that there are no type i coupons among the first k collected. Thus, with $S = \sum_{i=1}^n X_i$

$$p = P(S > 0).$$

Let $q_i = 1 - p_i$, and set $\lambda_i = P(X_i = 1) = q_i^k$and $\lambda = \sum_{i=1}^m \lambda_i$. Noting that the conditional distribution of $N_1, \ldots, N_m$ given that $X_j = 1$ is that of a multinomial with k trials where the probability that a trial outcome is i is p_i/q_j for $i \neq j$ and is 0 for $i = j$, the CBSM is as follows:

1. Generate J such that $P(J = i) = \lambda_i/\lambda, \quad i = 1, \ldots, m$. Suppose the generated value of J is j.
2. Generate the multinomial vector $(N_1, \ldots, N_m)$ yielding the number of outcomes of each type when k independent trials are performed, where the

probability of a type i trial outcome is 0 when $i = j$ and is p_i/q_j when $i \neq j$.

3. Set $X_i = I\{N_i = 0\}, i = 1, \ldots, m$
4. Let $S^* = \sum_{i=1}^{m} X_i$ and return the unbiased estimator λ/S^*.

The following table gives the variances of the CBSM and the raw simulation estimator $I = I\{N > k\}$ of $P(N > k)$ for various values of k when $p_i = i/55, i = 1, \ldots, 10$.

k	$P(N > k)$	Var(I)	Var(λ/S^*)
50	0.54	0.25	0.026
100	0.18	0.15	0.00033
150	0.07	0.06	9×10^{-6}
200	0.03	0.03	1.6×10^{-7}

Moreover, the preceding estimator can be further improved by using both stratified sampling and a control variable. For instance, suppose we plan to do proportional stratification, setting $J = j$ in $r\lambda_j/\lambda$ of these runs. Then in those runs using $J = j$ we can use S^* as a control variable. Its conditional mean is

$$E[S^*|J = j] = \sum_{i=1}^{m} E[X_i|X_j = 1] = 1 + \sum_{i \neq j}(1 - \frac{p_i}{q_j})^k. \qquad \square$$

Example 10b Consider the model of Example 9b, which is concerned with a system composed of n independent components, and suppose that we want to estimate the probability that the system is failed, when this probability is very small. Now, for any system of the type considered in Example 9b there will always be a unique family of sets $\{C_1, \ldots, C_m\}$, none of which is a subset of another, such that the system will be failed if and only if all the components of at least one of these sets are failed. These sets are called the *minimal cut sets* of the system.

Let Y_j, $j = 1, \ldots, n$ equal 1 if component j is failed and let it equal 0 otherwise, and let $q_j = P\{Y_j = 1\}$ denote the probability that component j is failed. Now, for $i = 1, \ldots, m$, let

$$X_i = \prod_{j \in C_i} Y_j$$

That is, X_i is the indicator for the event that all components in C_i are failed. If we let $S = \sum_i X_i$, then θ, the probability that the system is failed, is given by

$$\theta = P\{S > 0\}$$

We will now show how to make use of the Conditional Bernoulli Sampling Method to efficiently estimate θ.

First, let $\lambda_i = E[X_i] = \prod_{j \in C_i} q_j$, and let $\lambda = \sum_i \lambda_i$. Now, simulate the value of J, a random variable that is equal to i with probability λ_i/λ, $i = 1, \ldots, m$. Then set Y_i equal to 1 for all $i \in C_J$, and simulate the value of all of the other Y_i, $i \notin C_j$, by letting them equal 1 with probability q_i and 0 otherwise. Let S^* denote the resulting number of minimal cut sets that have all their components down, and note that $S^* \geq 1$. It then follows that λ/S^* is an unbiased estimator of θ. Since $S^* \geq 1$, it also follows that

$$0 \leqslant \lambda/S^* \leqslant \lambda$$

and so when λ, the mean number of minimal cut sets that are down, is very small the estimator λ/S^* will have a very small variance.

For instance, consider a 3-of-5 system that fails if at least 3 of the 5 components are failed, and suppose that each component independently fails with probability q. For this system, the minimal cut sets will be the $\binom{5}{3} = 10$ subsets of size 3. Since all the component failures are the same, the value of I will play no role. Thus, the preceding estimate can be obtained by supposing that components 1, 2, and 3 are all failed and then generating the status of the other two. Thus, by considering the number of components 4 and 5 that are failed, it follows since $\lambda = 10q^3$ that the distribution of the estimator is

$$\begin{aligned}
P\{\lambda/S^* &= 10q^3\} = (1-q)^2 \\
P\{\lambda/S^* &= 10q^3/4\} = 2q(1-q) \\
P\{\lambda/S^* &= q^3\} = q^2
\end{aligned}$$

Hence, with $p = 1 - q$,

$$\begin{aligned}
\mathrm{Var}(\lambda/S^*) &= E[(\lambda/S^*)^2] - (E[\lambda/S^*])^2 \\
&= 100q^6[p^2 + pq/8 + q^2/100 - (p^2 + pq/2 + q^2/10)^2]
\end{aligned}$$

The following table gives the value of θ and the ratio of $\mathrm{Var}(R)$ to the variance of the estimator λ/S^* for a variety of values of q, where $\mathrm{Var}(R) = \theta(1-\theta)$ is the variance of the raw simulation estimator.

q	θ	Var(R)/Var(λ/S*)
0.001	9.985×10^{-9}	8.896×10^{10}
0.01	9.851×10^{-6}	8,958,905
0.1	0.00856	957.72
0.2	0.05792	62.59
0.3	0.16308	12.29

Thus, for small q, $\mathrm{Var}(\lambda/S^*)$ is roughly of the order θ^2, whereas $\mathrm{Var}(R) \approx \theta$. $\square$

Example 10c Waiting for a Pattern Let Y_i, $i \geq 1$, be a sequence of independent and identically distributed discrete random variables with probability mass function $P_j = P\{Y_i = j\}$. Let $i_1, \ldots, i_k$ be a fixed sequence of possible values of these random variables and define

$$N = \min\{i : i \geq k, Y_{i-j} = i_{k-j}, j = 0, 1, \ldots, k-1\}$$

That is, N is the first time the pattern $i_1, \ldots, i_k$ occurs. We are interested in using simulation to estimate $\theta = P\{N \leqslant n\}$, in cases where θ is small. Whereas the usual simulation estimator is obtained by simulating the sequence of random variables until either the pattern occurs or it is no longer possible for it to occur by time n (and letting the estimator for that run be 1 in the former case and 0 in the latter), we will show how the CBSM can be applied to obtain a more efficient simulation estimator.

To begin, let

$$X_i = 1 \quad \text{if } Y_i = i_k, Y_{i-1} = i_{k-1}, \ldots, Y_{i-k+1} = i_1$$

and let it be 0 otherwise. In other words, X_i is equal to 1 if the pattern occurs (not necessarily for the first time) at time i. Let

$$S = \sum_{i=k}^{n} X_i$$

denote the number of times the pattern has occurred by time n and note that

$$\theta = P\{N \leqslant n\} = P\{S > 0\}$$

Since, for $k \leqslant i \leqslant n$

$$\lambda_i = P\{X_i = 1\} = P_{i_1} P_{i_2} \cdots P_{i_k} \equiv p$$

it follows by the CBSM that

$$\theta = (n - k + 1) p E\left[\frac{1}{S} \middle| X_I = 1\right]$$

where I, independent of the Y_j, is equally likely to be any of the values $k, \ldots, n$. Thus, we can estimate θ by first simulating J, equally likely to be any of the values $k, \ldots, n$, and setting

$$Y_J = i_k, \qquad Y_{J-1} = i_{k-1}, \ldots, \qquad Y_{J-k+1} = i_1$$

We then simulate the other $n - k$ values Y_i according to the mass function P_j and let S^* denote the number of times the pattern occurs. The simulation estimator of θ from this run is

$$\hat{\theta} = \frac{(n - k + 1) p}{S^*}$$

For small values of $(n - k + 1) p$, the preceding will be a very efficient estimator of θ. □

10.2 Normalized Importance Sampling

Suppose we want to estimate $\theta = E[h(\mathbf{X})]$ where $\mathbf{X}$ is a random vector having density (or mass) function f. The importance sampling technique is to generate $\mathbf{X}$ from a density g having the property that $g(\mathbf{x}) = 0$ implies that $f(\mathbf{x}) = 0$, and then taking $h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})$as the estimator of θ. That this is an unbiased estimator follows from

$$\theta = E_f[h(\mathbf{X})] = \int h(\mathbf{x})f(\mathbf{x})d(\mathbf{x}) = \int h(\mathbf{x})\frac{f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d(\mathbf{x}) = E_g\left[h(\mathbf{X})\frac{f(\mathbf{X})}{g(\mathbf{X})}\right]$$

If we now generate k such vectors $\mathbf{X}_1, \ldots, \mathbf{X}_k$ from the density g then the importance sampling estimator of θ based on these runs, call it $\hat{\theta}_{im}$, is

$$\hat{\theta}_{im} = \frac{\sum_{i=1}^k h(\mathbf{X}_i)f(\mathbf{X}_i)/g(\mathbf{X}_i)}{k}$$

The *normalized importance sampling estimator* replaces the divisor k in the preceding by $\sum_{i=1}^k f(\mathbf{X}_i)/g(\mathbf{X}_i)$. That is, the normalized importance sampling estimator, call it $\hat{\theta}_{nim}$, is

$$\hat{\theta}_{nim} = \frac{\sum_{i=1}^k h(\mathbf{X}_i)f(\mathbf{X}_i)/g(\mathbf{X}_i)}{\sum_{i=1}^k f(\mathbf{X}_i)/g(\mathbf{X}_i)}$$

Although $\hat{\theta}_{nim}$will not be an unbiased estimator of θ it will be a *consistent* estimator, meaning that with probability 1 it will converge to θ as the number of runs k goes to infinity. That this is true is seen by dividing its numerator and denominator by k to obtain

$$\hat{\theta}_{nim} = \frac{\frac{1}{k}\sum_{i=1}^k h(\mathbf{X}_i)f(\mathbf{X}_i)/g(\mathbf{X}_i)}{\frac{1}{k}\sum_{i=1}^k f(\mathbf{X}_i)/g(\mathbf{X}_i)}$$

Now, by the strong law of large numbers

$$\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^k h(\mathbf{X}_i)f(\mathbf{X}_i)/g(\mathbf{X}_i) = E_g[h(\mathbf{X})f(\mathbf{X})/g(\mathbf{X})] = E_f[h(\mathbf{X})] = \theta$$

and, again by the strong law of large numbers,

$$\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^k f(\mathbf{X}_i)/g(\mathbf{X}_i) = E_g[f(\mathbf{X})/g(\mathbf{X})] = \int\frac{f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d(\mathbf{x}) = \int f(\mathbf{x})d(\mathbf{x}) = 1$$

Hence, with probability 1, the numerator of $\hat{\theta}_{nim}$ converges to θ and the denominator converges to 1, showing that $\hat{\theta}_{nim}$ converges to θ as $k \to \infty$.

Remark We have previously seen the normalized importance sampling technique. Indeed, it is equivalent to the technique used to obtain the Eq. (9.14) estimator of $\theta = E_f[h(\mathbf{X})|\mathbf{X} \in \mathcal{A}]$. The estimator of (9.14) samples k random vectors according to the density g and then uses the estimate

$$\frac{\sum_{i=1}^{k} h(\mathbf{X}_i)I(\mathbf{X}_i \in \mathcal{A})f(\mathbf{X}_i)/g(\mathbf{X}_i)}{\sum_{i=1}^{k} I(\mathbf{X}_i \in \mathcal{A})f(\mathbf{X}_i)/g(\mathbf{X}_i)}$$

If we take $\mathcal{A}$ to be all of n-space then $I(\mathbf{X}_i \in \mathcal{A}) \equiv 1$, the problem becomes one of estimating $E_f[h(\mathbf{X})]$, and the preceding estimator is the normalized importance sampling estimator. □

An important feature of the normalized importance sampling estimator is that it can be utilized in cases where the density function f is only known up to a multiplicative constant. That is, for a known function $f_0(\mathbf{x})$ we may know that

$$f(\mathbf{x}) = Cf_0(\mathbf{x})$$

where $C^{-1} = \int f_0(\mathbf{x})\, d(\mathbf{x})$ may be difficult to compute. Because

$$\hat{\theta}_{nim} = \frac{\frac{1}{k}\sum_{i=1}^{k} h(\mathbf{X}_i)f_0(\mathbf{X}_i)/g(\mathbf{X}_i)}{\frac{1}{k}\sum_{i=1}^{k} f_0(\mathbf{X}_i)/g(\mathbf{X}_i)}$$

does not depend on the value of C, it can be used to estimate $\theta = E_f[h(\mathbf{X})]$ even when C is unknown.

Example 10d Let $X_i, i = 1, \ldots, r$, be independent binomial random variables with χ_i having parameters (n_i, p_i). Let $n = \sum_{i=1}^{r} n_i$ and $S = \sum_{i=1}^{r} X_i$, and suppose that we want to use simulation to estimate

$$\theta = E[h(X_1, \ldots, X_r)|S = m]$$

where h is a specified function and where $0 < m < n$. To start, suppose we determine the conditional probability mass function of $X_1, \ldots, X_r$ given that $S = m$. For $i_1, \ldots, i_r$ being nonnegative integers that sum to m we have

$$\begin{aligned} P(X_1 = i_1, \ldots, X_r = i_r|S = m) &= \frac{P(X_1 = i_1, \ldots, X_r = i_r)}{P(S = m)} \\ &= \frac{\prod_{j=1}^{r} \binom{n_j}{i_j} p_j^{i_j}(1 - p_j)^{n_j - i_j}}{P(S = m)} \end{aligned}$$

However, because the p_j need not be equal, it is difficult to compute $P(S = m)$. Thus, in essence the joint mass function under consideration is only known up to a multiplicative constant.

To get around this difficulty, let $Y_i, i = 1, \ldots, r$, be independent Binomial random variables with Y_i having parameters (n_i, p). Using that $S_y \equiv \sum_{i=1}^r Y_i$ is binomial with parameters (n, p), we see that for $\sum_{j=1}^r i_j = m$

$$P(Y_1 = i_1, \ldots, Y_r = i_r | S_y = m) = \frac{\prod_{j=1}^r \binom{n_j}{i_j} p^{i_j}(1-p)^{n_j - i_j}}{\binom{n}{m} p^m (1-p)^{n-m}}$$

$$= \frac{\binom{n_1}{i_1}\binom{n_2}{i_2}\cdots\binom{n_r}{i_r}}{\binom{n}{m}} \tag{10.3}$$

The conditional distribution of $Y_1, \ldots, Y_r$ given that their sum is m is, therefore, that of the numbers of balls of each of r types chosen when m balls are randomly chosen from an urn consisting of n balls, of which n_i are type i for each $i = 1, \ldots, r$. Consequently, given $\sum_{i=1}^r Y_i = m$, the Y_i can be generated sequentially, with all the conditional distributions being hypergeometric. That is, the conditional distribution of Y_j given that $S_y = m$ and $Y_i = y_i, i = 1, \ldots, j-1$ is that of a hypergeometric distributed as the number of red balls chosen when $m - \sum_{i=1}^{j-1} y_i$ balls are to be randomly chosen from an urn containing $\sum_{i=j}^r n_i$ balls of which n_j are red.

Now, if we let

$$R(i_1, \ldots, i_r) = \prod_{j=1}^r p_j^{i_j}(1-p_j)^{n_j - i_j}$$

then

$$\frac{P(X_1 = i_1, \ldots, X_r = i_r | S = m)}{P(Y_1 = i_1, \ldots, Y_r = i_r | S_y = m)} = \frac{\binom{n}{m}}{P(S = m)} R(i_1, \ldots, i_r)$$

Hence, we can estimate θ by generating k vectors $\mathbf{Y}_1, \ldots, \mathbf{Y}_k$ having the mass function (10.3), and then using the estimator

$$\hat{\theta}_{nim} = \frac{\sum_{i=1}^k h(\mathbf{Y}_i) R(\mathbf{Y}_i)}{\sum_{i=1}^k R(\mathbf{Y}_i)}$$ □

Example 10e Let $X_i, i = 1, \ldots, r$ be independent exponential random variables with rates $\lambda_i, i = 1, \ldots, r$, and suppose we want to estimate

$$\theta = E[h(X_1, \ldots, X_r) | S = t]$$

for some specified function h, where $S = \sum_{i=1}^{r} X_i$. To start, let us determine the conditional density function of $X_1, \ldots, X_{r-1}$ given that $S = t$. Calling this conditional density function f, we have for positive values $x_1, \ldots, x_{r-1}$ for which $\sum_{i=1}^{r-1} x_i < t$

$$\begin{aligned}
f(x_1, \ldots, x_r) &= f_{X_1,\ldots,X_{r-1}}(x_1, \ldots, x_{r-1}|S = t) \\
&= \frac{f_{X_1,\ldots,X_r}(x_1, \ldots, x_{r-1}, t - \sum_{i=1}^{r-1} x_i)}{f_S(t)} \\
&= \frac{\lambda_r e^{-\lambda_r (t - \sum_{i=1}^{r-1} x_i)} \prod_{i=1}^{r-1} \lambda_i e^{-\lambda_i x_i}}{f_S(t)} \\
&= \frac{e^{-\lambda_r t} e^{-\sum_{i=1}^{r-1}(\lambda_i - \lambda_r)x_i} \prod_{i=1}^{r} \lambda_i}{f_S(t)}
\end{aligned}$$

Now, with

$$h^*(x_1, \ldots, x_{r-1}) = h(x_1, \ldots, x_{r-1}, t - \sum_{i=1}^{r-1} x_i), \quad \sum_{i=1}^{r-1} x_i < t$$

it follows that

$$\theta = E[h(X_1, \ldots, X_r)|S = t] = E_f[h^*(X_1, \ldots, X_{r-1})]$$

However, because the λ_i need not be equal, it is difficult to compute $f_S(t)$ (which would be a gamma density if the λ_i were equal), and so the density function f is, in essence, only specified up to a multiplicative constant.

To make use of normalized importance sampling, let $U_1, \ldots, U_{r-1}$ be independent uniform random variables on $(0, t)$, and let $U_{(1)} < U_{(2)} < \ldots < U_{(r-1)}$ be their ordered values, Now, for $0 < y_1 < \ldots < y_{r-1} < t$, $U_{(i)}$ will equal y_i for all i if $U_1, \ldots, U_{r-1}$ is any of the $(r-1)!$ permutations of $y_1, \ldots, y_{r-1}$. Consequently,

$$f_{U_{(1)},\ldots,U_{(r-1)}}(y_1, \ldots, y_{r-1}) = \frac{(r-1)!}{t^{r-1}}, \quad 0 < y_1 < \ldots < y_{r-1} < t$$

If we now let

$$\begin{aligned}
X_1 &= U_{(1)} \\
X_i &= U_{(i)} - U_{(i-1)}, \ i = 2, \ldots, r-1
\end{aligned}$$

then it is easy to check (the Jacobian of the transformation is 1) that g, the joint density of $X_1, \ldots, X_{r-1}$, is

$$g(x_1, \ldots, x_{r-1}) = \frac{(r-1)!}{t^{r-1}}, \quad \sum_{i=1}^{r-1} x_i < t, \qquad \text{all} \qquad x_i > 0$$

It follows from the preceding that we can generate a random vector $X_1, \ldots, X_{r-1}$ having density g by generating $r - 1$ uniform $(0, t)$ random variables, ordering them, and then letting $X_1, \ldots, X_{r-1}$ be the differences of the successive ordered values.

Now, for $K = \frac{e^{-\lambda_r t} t^{r-1} \prod_{i=1}^{r} \lambda_i}{(r-1)! f_S(t)}$,

$$\frac{f(x_1, \ldots, x_{r-1})}{g(x_1, \ldots, x_{r-1})} = K e^{-\sum_{i=1}^{r-1} (\lambda_i - \lambda_r) x_i}$$

Consequently, if for $\mathbf{x} = (x_1, \ldots, x_{r-1})$, we define

$$R(\mathbf{x}) = e^{-\sum_{i=1}^{r-1} (\lambda_i - \lambda_r) x_i}$$

then we can estimate θ by generating k vectors $\mathbf{X}_1, \ldots, \mathbf{X}_k$ from the density g and then using the estimator

$$\hat{\theta}_{nim} = \frac{\sum_{i=1}^{k} h^*(\mathbf{X}_i) R(\mathbf{X}_i)}{\sum_{i=1}^{k} R(\mathbf{X}_i)} \qquad \square$$

10.3 Latin Hypercube Sampling

Suppose we wanted to use simulation to compute $\theta = E[h(U_1, \ldots, U_n)]$ where h is an arbitrary function and $U_1, \ldots, U_n$ are independent uniform (0, 1) random variables. That is, we want to compute

$$E[h(U_1, \ldots, U_n)] = \int_0^1 \int_0^1 \cdots \int_0^1 h(x_1, \ldots, x_n) dx_1 dx_2 \cdots dx_n$$

The standard approach would be to generate some number, say r, successive n vectors of independent uniform (0, 1) random variables:

$$\begin{aligned}
\mathbf{U}_1 &= (U_{1,1}, U_{1,2}, \ldots, U_{1,j}, \ldots, U_{1,n}) \\
\mathbf{U}_2 &= (U_{2,1}, U_{2,2}, \ldots, U_{2,j}, \ldots, U_{2,n}) \\
\ldots &= \ldots \\
\mathbf{U}_i &= (U_{i,1}, U_{i,2}, \ldots, U_{i,j}, \ldots, U_{i,n}) \\
\ldots &= \ldots \\
\mathbf{U}_r &= (U_{r,1}, U_{r,2}, \ldots, U_{r,j}, \ldots, U_{r,n})
\end{aligned}$$

then evaluate h at each of these vectors and use $\frac{1}{r} \sum_{i=1}^{r} h(\mathbf{U}_i)$ as the simulation estimator.

In the preceding, the values $U_{1,j}, U_{2,j}, \ldots, U_{r,j}$ taken for U_j in the r successive runs are independent and uniformly distributed on $(0, 1)$. Intuitively, a better approach would be to stratify these r values so that exactly one of them is in the interval $(\frac{k-1}{r}, \frac{k}{r})$ for each $k = 1, \ldots, r$. It is also intuitive that after doing this for each $j = 1, \ldots, n$ we would want to use the resulting nr values to make r n-vectors in a random manner so as to avoid such things as having one of the vectors consist of component values that are all uniformly distributed over $(0, \frac{1}{r})$, and so on. To accomplish this task note that if $p_1, \ldots, p_r$ is a permutation of $1, \ldots, r$ then

$$\frac{U_1 + p_1 - 1}{r}, \ldots, \frac{U_i + p_i - 1}{r}, \ldots, \frac{U_r + p_r - 1}{r}$$

is a sequence of r independent random variables, one being uniform on $(\frac{k-1}{r}, \frac{k}{r})$ for each $k = 1, \ldots, r$. Using this fact, we can construct our r n-vectors by first generating n independent random permutations of $1, \ldots, r$. Denoting these random permutations as $(\pi_{1,j}, \pi_{2,j}, \ldots, \pi_{r,j})$, $j = 1, \ldots, n$, and letting $U^*_{i,j} = \frac{U_{i,j}+\pi_{i,j}-1}{r}$, the r vectors are $\mathbf{U}^*{}_i = (U^*_{i,1}, U^*_{i,2}, \ldots, U^*_{i,n})$, $i = 1, \ldots, r$. Evaluating the function h at each of these vectors then yields the estimate of θ. That is, the estimate of θ is $\hat{\theta} = \frac{1}{r}\sum_{i=1}^{r} h(\mathbf{U}^*{}_i)$.

For instance, suppose that $n = 2, r = 3$. Then we start by generating the 3 vectors

$$\begin{aligned}
\mathbf{U}_1 &= (U_{1,1}, \quad U_{1,2}) \\
\mathbf{U}_2 &= (U_{2,1}, \quad U_{2,2}) \\
\mathbf{U}_3 &= (U_{3,1}, \quad U_{3,2})
\end{aligned}$$

Now we generate two random permutations of the values 1, 2, 3. Say they are (1, 3, 2) and (2, 3, 1). Then the resulting 3 vectors at which h is to be evaluated are

$$\begin{aligned}
\mathbf{U}^*{}_1 &= \left(\frac{U_{1,1}+0}{3}, \quad \frac{U_{1,2}+1}{3}\right) \\
\mathbf{U}^*{}_2 &= \left(\frac{U_{2,1}+2}{3}, \quad \frac{U_{2,2}+2}{3}\right) \\
\mathbf{U}^*{}_3 &= \left(\frac{U_{3,1}+1}{3}, \quad \frac{U_{3,2}+0}{3}\right)
\end{aligned}$$

It is easy to see (see Problem 8) that $U^*_{i,j}$ is uniformly distributed over (0, 1). Consequently, because $U^*_{i,1}, \ldots, U^*_{i,n}$ are independent, it follows that $E[\hat{\theta}] = \theta$. Although it is common that

$$\mathrm{Var}(\hat{\theta}) \leqslant \frac{\mathrm{Var}\,(h(U_1, \ldots, U_n))}{r}$$

this need not always be the case. It is, however, always true when h is a monotone function.

Exercises

1. Use the conditional Bernoulli sampling method to estimate the probability that the bridge structure given in Fig. 9.1 will fail if each of components 1, 2, 3 fail with probability 0.05 and each of components 4, 5 fail with probability 0.01. Assume that the component failure events are independent. Compare the variance of your estimator with that of the raw simulation estimator.
2. Estimate the additional variance reduction that would be obtained in Example 1 if one uses a post-stratification.
3. Estimate the additional variance reduction that would be obtained in Example 1 if one uses antithetic variables.
4. Use the conditional Bernoulli sampling method to estimate the probability that a run of 10 consecutive heads occurs within the first 50 flips of a fair coin. Compare the variance of your estimator with that of the raw simulation estimator.
5. Use the conditional Bernoulli sampling method to estimate the probability that the pattern HTHTH occurs within the first 20 flips of a coin that comes up heads with probability .3.
6. Show that the normalized importance sampling technique can be applied when both densities f and g are only known up to a multiplicative constant, provided that one is able to generate from g
7. Give a procedure for determining $E[X]$ when X has density function
$$f(x) = Ce^{x+x^2}, \quad 0 < x < 1$$
8. If U is uniform on (0, 1) and π is equally likely to be any of $1, \ldots, r$, show that $\frac{U+\pi-1}{r}$ is uniform on (0, 1).
9. Let $\theta = E[e^{\sum_{i=1}^{10} U_i}]$, where $U_1, \ldots, U_{10}$ are independent uniform (0, 1) random variables.
   (a) Estimate θ by using a raw simulation estimator based on 100 runs. That is, generate 100 independent sets of 10 random numbers and take the average of the 100 resulting values of e raised to the sum of the 10 uniforms in each run. Compare your estimate with the actual value $\theta = (e-1)^{10} = 224.359$. Note how long it took the simulation to run.
   (b) Repeat part (a) this time using the Latin hypercube procedure.
   (c) Repeat parts (a) and (b) using different random numbers.
   (d) Does the Latin hypercube procedure appear to yield an improvement over raw simulation?
   (e) What other variance reduction ideas could be used?
10. In the Latin hypercube sampling approach explain why it is only necessary to generate $n-1$, rather than n, random permutations.