Contents

8

Variance reduction

Monte Carlo integration typically has an error variance of the form σ^2/n. We get a better answer by sampling with a larger value of n, but the computing time grows with n. Sometimes we can find a way to reduce σ instead. To do this, we construct a new Monte Carlo problem with the same answer as our original one but with a lower σ. Methods to do this are known as variance reduction techniques.

The techniques can be placed into groups, though no taxonomy is quite perfect. First we will look at antithetic sampling, stratification, and common random numbers. These methods all improve efficiency by sampling the input values more strategically. Next we will consider conditioning and control variates. These methods take advantage of closed form solutions to problems similar to the given one.

The last major method is importance sampling. Like some of the other methods, importance sampling also changes where we take the sample values, but rather than distributing them in more balanced ways it purposely oversamples from some regions and then corrects for this distortion by reweighting. It is thus a more radical reformulation of the problem and can be tricky to do well. We devote Chapter 9 to importance sampling. Some more advanced methods of variance reduction are given in Chapter 10.

8.1 Overview of variance reduction

Variance reductions are used to improve the efficiency of Monte Carlo methods. Before looking at individual methods, we discuss how to measure efficiency. Then we introduce some of the notation we need.

Measuring efficiency

Methods of variance reduction can sometimes bring enormous improvements compared to plain Monte Carlo. It is not uncommon for the value σ^2 to be reduced many thousand fold. It is also possible for a variance reduction technique to bring a very modest improvement, perhaps equivalent to reducing σ^2 by only 10%. What is worse, some methods will raise σ^2 in unfavorable circumstances.

The value of a variance reduction depends on more than the change in σ^2. It also depends on the computer's running time, possibly the memory consumed, and quite importantly, the human time taken to program and test the code.

Suppose for simplicity, that a baseline method is unbiased and estimates the desired quantity with variance σ_0^2/n, at a cost of nc_0, when n function evaluations are used. To get an error variance of τ^2 we need $n = \sigma_0^2/\tau^2$ and this will cost $c_0\sigma_0^2/\tau^2$. Here we are assuming that cost is measured in time and that overhead cost is small.

If an alternative unbiased method has variance σ_1^2/n and cost nc_1 under these conditions then it will cost us $c_1\sigma_1^2/\tau^2$ to achieve the same error variance τ^2 that the baseline method achieved. The efficiency of the new method, relative to the standard method is

$$E = \frac{c_0\sigma_0^2}{c_1\sigma_1^2}. \tag{8.1}$$

At any fixed level of accuracy, the old method takes E times as much work as the new one.

The efficiency has two factors, σ_0^2/σ_1^2 and c_0/c_1. The first is a mathematical property of the two methods that we can often handle theoretically. The second is more complicated. It can depend heavily on the algorithms used for each method. It can also depend on details of the computing environment, including the computer hardware, operating system, and implementation language. Numerical results for c_0/c_1 obtained in one setting do not necessarily apply to another.

There is no fixed rule for how large an efficiency improvement must be to make it worth using. In some settings, such as rendering computer graphics for animated motion pictures, where thousands of CPUs are kept busy for months, a 10% improvement (i.e., $E = 1.1$) brings meaningful savings. In other settings, such as a one-off computation, a 60-fold gain (i.e., $E = 60$) which turns a one minute wait into a one second wait, may not justify the cost of programming a more complicated method.

Computation costs so much less than human effort that we ordinarily require large efficiency gains to offset the time spent programming up a variance reduction. The impetus to seek out an efficiency improvement may only come when we find ourselves waiting a very long time for a result, as for example, when we need to place our entire Monte Carlo calculation within a loop representing many variants of the problem. A very slow computation costs more than just the computer's time. It may waste time for those waiting for the answer. Also, slow computations reduce the number of alternatives that one can explore.

The efficiency gain necessary to justify using a method is less if the programming effort can be amortized over many applications. The threshold is high for a one time program, lower for something that we are adding to our personal library, lower still for code to share with a few coworkers and even lower for code to be put into a library or simulation tool for general use.

In the numerical examples in this chapter, some of the methods achieve quite large efficiency gains, while others are more modest. These results should not be taken as inherent to the methods. All of the methods are capable of a great range of efficiency improvements.

Notation

Monte Carlo problems can be formulated through expectations or integrals or for discrete random variables, as sums. Generally, we will pick whichever format makes a given problem easiest to work with.

We suppose that the original Monte Carlo problem is to find $\mu = \mathbb{E}(f(\boldsymbol{X}))$ where $\boldsymbol{X}$ is a random variable from the set $\mathcal{D} \subset \mathbb{R}^d$ with distribution p. When p is a probability density function we may write $\mu = \int_{\mathcal{D}} f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. Most of the time we just write $\mu = \int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ with the understanding that $p(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \notin \mathcal{D}$. The integral version is convenient when we are reparameterizing the problem. Then, following the rules for integration is the best way to be sure of getting the right answer.

Monte Carlo sampling of $\boldsymbol{X} \sim p$ is often based on s uniform random variables through a transformation $\boldsymbol{X} = \psi(\boldsymbol{U})$, for $\boldsymbol{U} \sim \mathbf{U}(0,1)^s$. Some variance reductions (e.g., antithetic sampling and stratification) are easier to apply directly to $\boldsymbol{U}$ rather than to $\boldsymbol{X}$. For this case we write $\mu = \int_{(0,1)^s} f(\psi(\boldsymbol{u}))\,\mathrm{d}\boldsymbol{u}$, or $\mu = \int_{(0,1)^s} f^*(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$, where $f^*(\boldsymbol{u}) = f(\psi(\boldsymbol{u}))$. When we don't have to keep track of both transformed and untransformed versions, then we just write $\mu = \int_{(0,1)^d} f(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$, subsuming ψ into f. This expression may be abbreviated to $\mu = \int f(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$ when the domain of $\boldsymbol{u}$ is clear from context.

Similar expressions hold for discrete random variables. Also some of the methods extend readily to $d = \infty$.

8.2 Antithetics

When we are using Monte Carlo averages of quantities $f(\boldsymbol{X}_i)$ then the randomness in the algorithm leads to some error cancellation. In antithetic sampling we try to get even more cancellation. An antithetic sample is one that somehow gives the opposite value of $f(\boldsymbol{x})$, being low when $f(\boldsymbol{x})$ is high and vice versa. Ordinarily we get an opposite f by sampling at a point $\widetilde{\boldsymbol{x}}$ that is somehow opposite to $\boldsymbol{x}$.

Let $\mu = \mathbb{E}(\boldsymbol{X})$ for $\boldsymbol{X} \sim p$, where p is a symmetric density on the symmetric set $\mathcal{D}$. Here, symmetry is with respect to reflection through the center point $\boldsymbol{c}$ of $\mathcal{D}$. If we reflect $\boldsymbol{x} \in \mathcal{D}$ through $\boldsymbol{c}$ we get the point $\widetilde{\boldsymbol{x}}$ with $\widetilde{\boldsymbol{x}} - c = -(\boldsymbol{x} - c)$, that is $\widetilde{\boldsymbol{x}} = 2\boldsymbol{c} - \boldsymbol{x}$. Symmetry means that $p(\widetilde{\boldsymbol{x}}) = p(\boldsymbol{x})$ including the constraint

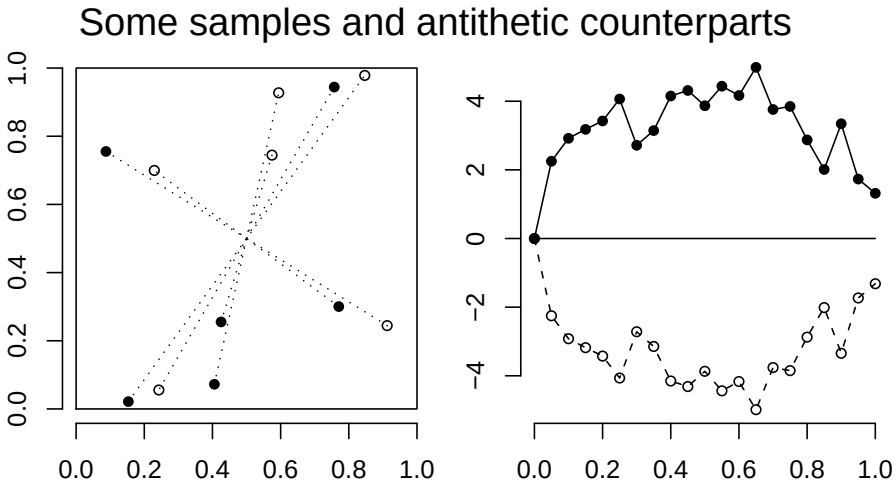

Figure 8.1: The left panel shows 6 points $\boldsymbol{u}_i \in [0,1]^2$ as solid points, connected to their antithetic counterparts $\widetilde{\boldsymbol{u}}_i = 1-\boldsymbol{u}_i$, shown as open circles. The right panel shows one random trajectory of 20 points joined by solid lines and connected to the origin, along with its antithetic mirror image in open points.

that $\boldsymbol{x} \in \mathcal{D}$ if and only if $\widetilde{\boldsymbol{x}} \in \mathcal{D}$. For basic examples, when p is $\mathcal{N}(0,\Sigma)$ then $\widetilde{\boldsymbol{x}} = -\boldsymbol{x}$, and when p is $\mathbf{U}(0,1)^d$ we have $\widetilde{\boldsymbol{x}} = 1-\boldsymbol{x}$ componentwise. The antithetic counterpart of a random curve could be its reflection in the horizontal axis. See Figure 8.1 for examples. From the symmetry it follows that $\widetilde{\widetilde{\boldsymbol{x}}} = \boldsymbol{x}$.

The ***antithetic sampling*** estimate of μ is

$$\hat{\mu}_{\text{anti}} = \frac{1}{n}\sum_{i=1}^{n/2}\bigl(f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i)\bigr), \tag{8.2}$$

where $\boldsymbol{X}_i \overset{\text{iid}}{\sim} p$, and n is an even number.

The rationale for antithetic sampling is that each value of $\boldsymbol{x}$ is balanced by its opposite $\widetilde{\boldsymbol{x}}$ satisfying $(\boldsymbol{x}+\widetilde{\boldsymbol{x}})/2 = \boldsymbol{c}$. Whether this balance is helpful depends on f. Clearly if f is nearly linear we could obtain a large improvement. Suppose that $\sigma^2 = \mathbb{E}((f(\boldsymbol{X})-\mu)^2) < \infty$. Then the variance in antithetic sampling is

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_{\text{anti}}) &= \mathrm{Var}\biggl(\frac{1}{n}\sum_{i=1}^{n/2} f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i)\biggr)\\
&= \frac{n/2}{n^2}\,\mathrm{Var}(f(\boldsymbol{X}) + f(\widetilde{\boldsymbol{X}}))\\
&= \frac{1}{2n}\Bigl(\mathrm{Var}(f(\boldsymbol{X})) + \mathrm{Var}(f(\widetilde{\boldsymbol{X}})) + 2\mathrm{Cov}(f(\boldsymbol{X}), f(\widetilde{\boldsymbol{X}}))\Bigr)
\end{aligned}$$

$$= \frac{\sigma^2}{n}(1+\rho) \tag{8.3}$$

where $\rho = \mathrm{Corr}(f(\boldsymbol{X}), f(\widetilde{\boldsymbol{X}}))$.

From $-1 \leqslant \rho \leqslant 1$ we obtain $0 \leqslant \sigma^2(1+\rho) \leqslant 2\sigma^2$. In the best case, antithetic sampling gives the exact answer from just one pair of function evaluations. In the worst case it doubles the variance. Both cases do arise.

It is clear that a negative correlation is favorable. If f happens to be monotone in all d components of $\boldsymbol{x}$, then it is known that $\rho < 0$. Monotonicity of f is a safe harbor: if f is monotone then we're sure antithetic sampling will reduce the variance. We can often establish monotonicity theoretically, for example by differentiating f. But $\rho < 0$ can hold without f being monotone in any of its inputs. Conversely ρ can be just barely negative when f is monotone. As a result, monotonicity alone is not a good guide to whether antithetic sampling will bring a large gain. See Exercise 8.1.

To get a qualitative understanding of antithetic sampling, break f into even and odd parts via

$$\begin{aligned} f(\boldsymbol{x}) &= \frac{f(\boldsymbol{x}) + f(\widetilde{\boldsymbol{x}})}{2} + \frac{f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})}{2} \\ &\equiv f_{\mathrm{E}}(\boldsymbol{x}) + f_{\mathrm{O}}(\boldsymbol{x}). \end{aligned}$$

The even part satisfies $f_{\mathrm{E}}(\boldsymbol{x}) = f_{\mathrm{E}}(\widetilde{\boldsymbol{x}})$ and $\int_{\mathcal{D}} f_{\mathrm{E}}(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu$. The odd part satisfies $f_{\mathrm{O}}(\boldsymbol{x}) = -f_{\mathrm{O}}(\widetilde{\boldsymbol{x}})$ and $\int_{\mathcal{D}} f_{\mathrm{O}}(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0$.

The even and odd parts of f are orthogonal. This is not a surprise, because the product $f_{\mathrm{O}}(\boldsymbol{x})f_{\mathrm{E}}(\boldsymbol{x})$ is an odd function. But to be careful and rule out $\mathbb{E}(|f_{\mathrm{O}}(\boldsymbol{X})f_{\mathrm{E}}(\boldsymbol{X})|) = \infty$, we compute directly that

$$\begin{aligned} \int_{\mathcal{D}} f_{\mathrm{E}}(\boldsymbol{x})f_{\mathrm{O}}(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} &= \int_{\mathcal{D}} \Big(\frac{f(\boldsymbol{x}) + f(\widetilde{\boldsymbol{x}})}{2}\Big)\Big(\frac{f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})}{2}\Big)p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{4}\int_{\mathcal{D}} \big(f(\boldsymbol{x})^2 - f(\widetilde{\boldsymbol{x}})^2\big)p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0. \end{aligned}$$

Now it follows easily that $\sigma^2 = \sigma^2_{\mathrm{E}} + \sigma^2_{\mathrm{O}}$ where $\sigma^2_{\mathrm{E}} = \int_{\mathcal{D}}(f_{\mathrm{E}}(\boldsymbol{x}) - \mu)^2 p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and $\sigma^2_{\mathrm{O}} = \int_{\mathcal{D}} f_{\mathrm{O}}(\boldsymbol{x})^2 p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$.

Reworking equation (8.3) yields $\hat{\mu}_{\mathrm{anti}} = (2/n)\sum_{i=1}^{n/2} f_{\mathrm{E}}(\boldsymbol{X}_i)$. Therefore $\mathrm{Var}(\hat{\mu}_{\mathrm{anti}}) = 2\sigma^2_{\mathrm{E}}/n$ and we can combine this with the variance of ordinary Monte Carlo sampling as follows:

$$\begin{pmatrix} V(\hat{\mu}) \\ V(\hat{\mu}_{\mathrm{anti}}) \end{pmatrix} = \frac{1}{n}\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}\begin{pmatrix} \sigma^2_{\mathrm{E}} \\ \sigma^2_{\mathrm{O}} \end{pmatrix}. \tag{8.4}$$

We see from (8.4) that antithetic sampling eliminates the variance contribution of f_{O} but doubles the contribution from f_{E}. Antithetic sampling is extremely beneficial for integrands that are primarily odd functions of their inputs, having $\sigma^2_{\mathrm{O}} \gg \sigma^2_{\mathrm{E}}$. The connection to correlation is via $\rho = (\sigma^2_{\mathrm{E}} - \sigma^2_{\mathrm{O}})/(\sigma^2_{\mathrm{E}} + \sigma^2_{\mathrm{O}})$ (Exercise 8.3).

The analysis above shows that antithetic sampling reduces variance if $\rho = \mathrm{Corr}(f(\boldsymbol{X}), f(\widetilde{\boldsymbol{X}})) < 0$, or equivalently, if $\sigma^2_{\mathrm{O}} > \sigma^2_{\mathrm{E}}$. That analysis is appropriate when the most of the computation is in evaluating f and there is no economy in evaluating both $f(\boldsymbol{X})$ and $f(\widetilde{\boldsymbol{X}})$.

Variance reduction is only part of the story because the cost of antithetic sampling using n points could well be smaller than the cost of plain Monte Carlo with n points. That will happen if it is expensive to generate $\boldsymbol{X}$, compared to the cost of computing f, but inexpensive to generate $\widetilde{\boldsymbol{X}}$. For example, $\boldsymbol{X}$ might be a carefully constructed and expensive sample path from a Gaussian process while $\widetilde{\boldsymbol{X}} = -\boldsymbol{X}$.

We can explore this effect by letting c_x be the cost of generating $\boldsymbol{X}$, and c_f be the cost of computing $f(\boldsymbol{X})$ once we have $\boldsymbol{X}$. We also let $\widetilde{c}_x$ and $\widetilde{c}_f$ be the corresponding costs for the antithetic sample. For illustration, suppose that to a reasonable approximation $\widetilde{c}_x = 0$ and $\widetilde{c}_f = c_f$. In special circumstances $\widetilde{c}_f < c_f$ because it may be possible to reuse some computation.

Under the assumptions we are exploring, the efficiency of antithetic sampling relative to plain Monte Carlo is

$$E_{\mathrm{anti}} = \frac{2c_x + 2c_f}{c_x + 2c_f} \times \frac{\sigma^2_{\mathrm{O}} + \sigma^2_{\mathrm{E}}}{2\sigma^2_{\mathrm{E}}}.$$

Then antithetic sampling is more efficient than plain Monte Carlo if

$$\frac{\sigma^2_{\mathrm{O}}}{\sigma^2_{\mathrm{E}}} > \frac{c_f}{c_x + c_f}.$$

If generating $\boldsymbol{x}$ costs ten times as much as computing f then antithetic sampling pays off when $\sigma^2_{\mathrm{O}}/\sigma^2_{\mathrm{E}} > 1/11$.

Because antithetic samples have dependent values within pairs, the usual variance estimate must be modified. The most straightforward approach is to analyze the data as a sample of size $m = n/2$ values of $f_{\mathrm{E}}(\boldsymbol{X})$. Let $Y_i = f_{\mathrm{E}}(\boldsymbol{X}_i) = (f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i))/2$ for $i = 1, \ldots, m = n/2$. Then take

$$\hat{\mu}_{\mathrm{anti}} = \frac{1}{m}\sum_{i=1}^{m} Y_i, \quad \text{and}$$

$$s^2_{\mathrm{anti}} = \frac{1}{m-1}\sum_{i=1}^{m} (Y_i - \hat{\mu}_{\mathrm{anti}})^2,$$

and use s^2_{anti}/m as the estimate of $\mathrm{Var}(\hat{\mu}_{\mathrm{anti}})$.

8.3 Example: expected log return

As an example of antithetic sampling we consider the expected logarithmic return of a portfolio. There are K stocks and the portfolio has proportion

$\lambda_k \geqslant 0$ invested in stock k for $k = 1, \ldots, K$, with $\sum_{k=1}^K \lambda_k = 1$. The expected logarithmic return is

$$\mu(\lambda) = \mathbb{E}\big(\log\big(\textstyle\sum_{k=1}^K \lambda_k e^{X_k}\big)\big) \tag{8.5}$$

where $\boldsymbol{X} \in \mathbb{R}^K$ is the vector of returns. At the end of the time period, the allocations are proportional to $\lambda_k e^{X_k}$. By selling some of the stocks with the largest X_k and buying some with the smallest X_k, it is possible to rebalance the portfolio so that the fraction of value in stock k is once again λ_k.

The expected logarithmic return is interesting because if one keeps reinvesting and rebalancing the portfolio at N regular time intervals then, by the law of large numbers, one's fortune grows as $\exp(N\mu + o_p(N))$, assuming of course that vectors $\boldsymbol{X}$ for each time period are independent and identically distributed. See Luenberger (1998, Chapter 15). The log–optimal choice λ is the allocation that maximizes μ. Log–optimal portfolios are of interest to very long term investors. Luenberger (1998) describes other criteria as well.

Finding a model for the distribution of $\boldsymbol{X}$ and then choosing λ are challenging problems, but to illustrate antithetic sampling, simplified choices serve as well as elaborate ones. We focus on the problem of evaluating $\mu(\lambda)$ for a given λ. We probably have to solve that problem en route to finding the best λ and definitely need to solve it once we have chosen λ. Here we take $\lambda_k = 1/K$ for $k = 1, \ldots, K$ with $K = 500$. We also suppose that each marginal distribution is $X_k \sim \mathcal{N}(\delta, \sigma^2)$ but that $\boldsymbol{X}$ has the $\mathrm{t}(0, \nu, \Sigma)$ copula. Here $\delta = 0.001$, $\sigma = 0.03$, $\nu = 4$ and $\Sigma = \rho \mathbf{1}_K \mathbf{1}_K^{\mathsf{T}} + (1-\rho) I_K$ for $\rho = 0.3$. These values of δ and σ are chosen to reflect roughly a one week time frame.

Letting $f(\boldsymbol{X}) = \log(\sum_{k=1}^K e^{X_k}/K)$, the plain Monte Carlo estimate of μ is $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{X}_i)$. The antithetic counterpart to $\boldsymbol{X}_i$ has $\widetilde{X}_{ik} = 2\delta - X_{ik}$. Using $n = 10{,}000$ sample values we find $\hat{\rho}(f(\boldsymbol{X}), f(\widetilde{\boldsymbol{X}})) \doteq -0.999508$ and so the variance reduction factor from antithetic sampling is $(1+\rho)^{-1} \doteq 2030.0$.

For those $n = 10{,}000$ pairs we let $Y_i = (f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i))/2 = f_{\mathrm{E}}(\boldsymbol{X}_i)$ and get the estimate

$$\hat{\mu}_{\mathrm{anti}} = \frac{1}{n}\sum_{i=1}^n Y_i \doteq 0.00132.$$

The standard deviation is $s = \big((n-1)^{-1}\sum_{i=1}^n (Y_i - \hat{\mu}_{\mathrm{anti}})^2\big)^{1/2} \doteq 0.000252$. The 99% confidence interval for μ is

$$\hat{\mu}_{\mathrm{anti}} \pm 2.58 s n^{-1/2} \doteq 0.00132 \pm 6.49 \times 10^{-6}.$$

Antithetic sampling worked so well here because the function is nearly linear. The exponentials in (8.5) operate on a random variable that is usually near 0 and the logarithm operates on an argument that is usually near 1, and as a result the random variable whose expectation we take is nearly linear in $\boldsymbol{X}$. This near linearity is not limited to the particular λ and Σ we have used.

When $\boldsymbol{X}$ varies more widely, then the curvature of the exponential and logarithmic functions makes more of a difference and antithetic sampling will

Stocks	Period	Correlation	Reduction	Estimate	Uncertainty
20	week	−0.99957	2320.0	0.00130	6.35×10^{-6}
500	week	−0.99951	2030.0	0.00132	6.49×10^{-6}
20	year	−0.97813	45.7	0.06752	3.27×10^{-4}
500	year	−0.99512	40.2	0.06850	3.33×10^{-4}

Table 8.1: This table summarizes the results of the antithetic sampling to estimate the expected log return of a portfolio, as described in the text. The first column has the number K of stocks. The second column indicates whether the return was for a week or a year. The third column is the correlation between log returns and their antithetic counterpart. The fourth column turns this correlation into a variance reduction factor. Then comes the estimate of expected log return and the half width of a 99% confidence interval.

lose some effectiveness. Let's consider for example, annual rebalancing, and take $\delta = 52 \times 0.01$ and $\sigma = \sqrt{52} \times 0.03$. The annualized $\boldsymbol{X}$ has the same mean and variance as the sum of 52 IID copies of the weekly random variable. It does not have quite the same copula. We ignore that small difference and simulate using the same t copula as before. In this case, we find a reduced but still substantial variance reduction of about 40 fold. Conversely, running an example with $K = 20$ instead of 500 leads to a slightly bigger advantage for antithetic sampling. Four cases are summarized in Table 8.1.

8.4 Stratification

The idea in stratified sampling is to split up the domain $\mathcal{D}$ of $\boldsymbol{X}$ into separate regions, take a sample of points from each such region, and combine the results to estimate $\mathbb{E}(f(\boldsymbol{X}))$. Intuitively, if each region gets its fair share of points then we should get a better answer. Figure 8.2 shows two small examples of stratified domains. We might be able to do better still by oversampling within the important strata and undersampling those in which f is nearly constant.

We begin with the notation for stratified sampling. Then we show that stratified sampling is unbiased, find the variance of stratified sampling and show how to estimate that variance.

Our goal is to estimate $\mu = \int_{\mathcal{D}} f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. We partition $\mathcal{D}$ into mutually exclusive and exhaustive regions $\mathcal{D}_j$, for $j = 1, \ldots, J$. These regions are the strata. We write $\omega_j = \mathbb{P}(\boldsymbol{X} \in \mathcal{D}_j)$ and to avoid trivial issues, we assume $\omega_j > 0$. Next let $p_j(\boldsymbol{x}) = \omega_j^{-1} p(\boldsymbol{x}) \mathbb{1}_{\boldsymbol{x} \in \mathcal{D}_j}$, the conditional density of $\boldsymbol{X}$ given that $\boldsymbol{X} \in \mathcal{D}_j$.

To use stratified sampling, we must know the sizes ω_j of the strata, and we must also know how to sample $\boldsymbol{X} \sim p_j$ for $j = 1, \ldots, J$. These conditions are quite reasonable. When we are defining strata, we naturally prefer ones we can sample from. If however, we know ω_j but are unable to sample from p_j, then the method of post-stratification described on page 12 is available.

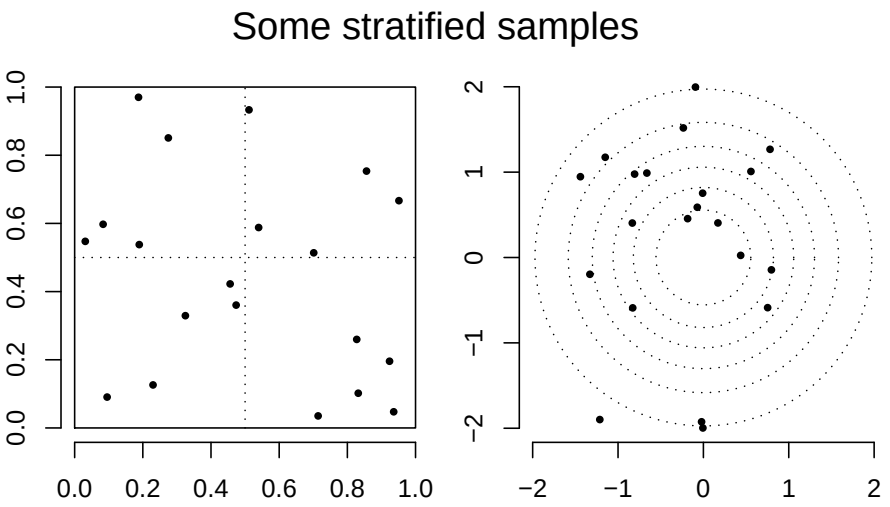

Figure 8.2: The left panel shows 20 points $\boldsymbol{x}_i \in [0,1]^2$ of which 5 are sampled uniformly from within each of four quadrants. The right panel shows 21 points from the $\mathcal{N}(0, I_2)$ distribution. There are 6 concentric rings separating the distribution into 7 equally probable strata. Each stratum has 3 points sampled from within it.

Let $\boldsymbol{X}_{ij} \sim p_j$ for $i = 1, \dots, n_j$ and $j = 1, \dots, J$ be sampled independently. The stratified sampling estimate of μ is

$$\hat{\mu}_{\text{strat}} = \sum_{j=1}^{J} \frac{\omega_j}{n_j} \sum_{i=1}^{n_j} f(\boldsymbol{X}_{ij}). \tag{8.6}$$

We choose $n_j > 0$ so that $\hat{\mu}_{\text{strat}}$ is properly defined. Unless otherwise specified, we make sure that $n_j \geqslant 2$, which will allow the variance estimate (8.10) below to be applied.

Now

$$\begin{aligned}\mathbb{E}(\hat{\mu}_{\text{strat}}) &= \sum_{j=1}^{J} \omega_j \mathbb{E}\biggl(\frac{1}{n_j}\sum_{i=1}^{n_j} f(\boldsymbol{X}_{ij})\biggr) = \sum_{j=1}^{J} \omega_j \int_{\mathcal{D}_j} f(\boldsymbol{x}) p_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \sum_{j=1}^{J} \int_{\mathcal{D}_j} f(\boldsymbol{x}) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{\mathcal{D}} f(\boldsymbol{x}) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu, \end{aligned} \tag{8.7}$$

and so stratified sampling is unbiased.

We study the variance of $\hat{\mu}_{\text{strat}}$ to determine when stratification is advantageous, and to see how to design an effective stratification. Let $\mu_j = \int_{\mathcal{D}_j} f(\boldsymbol{x}) p_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and $\sigma_j^2 = \int_{\mathcal{D}_j} (f(\boldsymbol{x}) - \mu_j)^2 p_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ be the j'th stratum mean

and variance, respectively. The variance of the stratified sampling estimate is

$$\mathrm{Var}(\hat{\mu}_{\mathrm{strat}}) = \sum_{j=1}^{J} \omega_j^2 \frac{\sigma_j^2}{n_j}. \tag{8.8}$$

An immediate consequence of (8.8) is that $\mathrm{Var}(\hat{\mu}_{\mathrm{strat}}) = 0$ for integrands f that are constant within strata $\mathcal{D}_j$. The variance of $f(\boldsymbol{X})$ can be decomposed into within– and between–stratum components as follows

$$\sigma^2 = \sum_{j=1}^{J} \omega_j \sigma_j^2 + \sum_{j=1}^{J} \omega_j (\mu_j - \mu)^2. \tag{8.9}$$

Equation (8.9) is simply $\mathrm{Var}(f(\boldsymbol{X})) = \mathbb{E}(\mathrm{Var}(f(\boldsymbol{X} \mid Z))) + \mathrm{Var}(\mathbb{E}(f(\boldsymbol{X} \mid Z)))$ where $Z \in \{1, \dots, J\}$ is the stratum containing the random point $\boldsymbol{X}$.

For error estimation, we write

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{ij}, \qquad s_j^2 = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (Y_{ij} - \hat{\mu}_j)^2, \quad \text{and}$$

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat}}) = \sum_{j=1}^{J} \omega_j^2 \frac{s_j^2}{n_j}. \tag{8.10}$$

Clearly $\mathbb{E}(s_j^2) = \sigma_j^2$ and so $\mathbb{E}(\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat}})) = \mathrm{Var}(\hat{\mu}_{\mathrm{strat}})$. A central limit theorem based 99% confidence interval for μ is

$$\hat{\mu}_{\mathrm{strat}} \pm 2.58 \sqrt{\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat}})}. \tag{8.11}$$

The CLT-based interval (8.11) is reasonable if all the n_j are large enough that each $\hat{\mu}_j$ is nearly normally distributed. This condition is sufficient but not necessary. The estimate $\hat{\mu}_{\mathrm{strat}}$ is a sum of J terms $\omega_j \hat{\mu}_j$. Even if every $n_j = 2$, it might be reasonable to apply a central limit theorem holding as $J \to \infty$ as described in Karr (1993, Chapter 7).

If we know ω_j but prefer not to sample $\boldsymbol{X} \sim p_j$ (or if we cannot do that), then we may still use the strata. In ***post-stratification*** we sample $\boldsymbol{X}_i \sim p$ and assign $\boldsymbol{X}_i$ to their strata after the fact. We let n_j be the number of sample points $\boldsymbol{X}_i \in \mathcal{D}_j$, let $\hat{\mu}_j$ be the average of $f(\boldsymbol{X}_i)$ for those points and s_j^2 be their sample variance. Then we estimate μ by the same $\hat{\mu}_{\mathrm{strat}}$ in (8.8) and use the same confidence interval (8.11) as before.

The main difference is that n_j are now random. There is also a risk of getting some $n_j = 0$ in which case we cannot actually compute $\hat{\mu}_{\mathrm{strat}}$ by (8.7). However $\mathbb{P}(\min_j n_j = 0) \leqslant \sum_{j=1}^{J} (1 - \omega_j)^n$ which we can make negligible by choosing n and the strata appropriately. Similarly, a sound choice for n and the strata $\mathcal{D}_j$ will make $n_j < 2$ very improbable.

Post-stratified sampling is a special case of the method of control variates. We will see this in Example 8.4 of §8.9.

A natural choice for stratum sample sizes is ***proportional allocation***, $n_j = n\omega_j$. In our analysis, we'll suppose that all the n_j are integers. We can usually choose n and $\mathcal{D}_j$ to make this so, or else accept small non-proportionalities due to rounding.

For proportional allocation, equation (8.6) for $\hat{\mu}_{\text{strat}}$ reduces to the ordinary sample mean

$$\hat{\mu}_{\text{prop}} = \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j} f(\boldsymbol{X}_{ij}). \tag{8.12}$$

Also, with proportional allocation, equation (8.8) for $\mathrm{Var}(\hat{\mu}_{\text{strat}})$ becomes

$$\sum_{j=1}^{J}\omega_j^2\frac{\sigma_j^2}{n\omega_j} = \frac{1}{n}\sum_{j=1}^{J}\omega_j\sigma_j^2. \tag{8.13}$$

Equation (8.13) allows us to show that stratified sampling with proportional allocation cannot have larger variance than ordinary MC sampling. Let $\sigma^2_{\text{W}} = \sum_{j=1}^{J}\omega_j\sigma_j^2$ and $\sigma^2_{\text{B}} = \sum_{j=1}^{J}\omega_j(\mu_j-\mu)^2$ be the within– and between–stratum variances. We can compare IID and proportional stratification in one equation:

$$\begin{pmatrix}\mathrm{Var}(\hat{\mu})\\ \mathrm{Var}(\hat{\mu}_{\text{prop}})\end{pmatrix} = \frac{1}{n}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}\sigma^2_{\text{B}}\\ \sigma^2_{\text{W}}\end{pmatrix}. \tag{8.14}$$

A good stratification scheme is one that reduces the within–stratum variance, ideally leaving $\sigma^2_{\text{B}} \gg \sigma^2_{\text{W}}$. If sampling from p_j is slower than sampling from p, then that reduces any efficiency gain from stratification.

Another way to look at proportional allocation is to construct the piece-wise constant function $h(\boldsymbol{x})$ with $h(\boldsymbol{x}) = \mu_j$ when $\boldsymbol{x} \in \mathcal{D}_j$. Then (Exercise 8.5),

$$\mathrm{Var}(\hat{\mu}_{\text{prop}}) = (1-\rho^2)\mathrm{Var}(\hat{\mu}), \tag{8.15}$$

where ρ is the correlation between $f(\boldsymbol{X})$ and $h(\boldsymbol{X})$ for $\boldsymbol{X} \sim p$.

A proportional allocation is not necessarily the most efficient. For instance, given two strata with equal ω_j but unequal σ_j^2, we benefit by taking fewer points from the less variable stratum. In the extreme, if $\sigma_j = 0$ then $n_j = 1$ is enough to tell us μ_j.

The problem of optimal sample allocation to strata has been solved in the survey sampling literature. The result is known as the Neyman allocation, and the formulation allows for unequal sampling costs from the different strata. Suppose that for unit costs $c_j > 0$ the stratified sampling costs $C + \sum_{j=1}^{J} n_j c_j$ to generate random variables and evaluate f. Here $C \geqslant 0$ is an overhead cost and c_j is the (expected) cost to generate $\boldsymbol{X}$ from p_j and then compute $f(\boldsymbol{X})$. To minimize variance subject to an upper bound on cost, take

$$n_j \propto \frac{\omega_j\sigma_j}{\sqrt{c_j}}. \tag{8.16}$$

The solution (8.16) also minimizes cost subject to a lower bound on variance. Equation (8.16) can be established by the method of Lagrange multipliers. These optimal values n_j usually need to be rounded to integers and some may have to be raised, if for other reasons we insist that all n_j be above some minimum such as 2.

When the sampling cost c_j is the same in every stratum then the optimal allocation has

$$n_j = \frac{n\omega_j\sigma_j}{\sum_{k=1}^{J}\omega_k\sigma_k}. \tag{8.17}$$

Let $\hat{\mu}_{n\text{–opt}}$ be the stratified sampling estimate (8.6) with optimal n_j from (8.17). By substituting (8.17) into the stratified sampling variance (8.8) we find that

$$\mathrm{Var}(\hat{\mu}_{n\text{–opt}}) = \frac{1}{n}\biggl(\sum_{j=1}^{J}\omega_j\sigma_j\biggr)^2 \leqslant \frac{1}{n}\sum_{j=1}^{J}\omega_j\sigma_j^2 = \mathrm{Var}(\hat{\mu}_{\mathrm{prop}}). \tag{8.18}$$

Equality holds in (8.18), only when σ_j is constant in j.

In typical applications, the values of σ_j are not known. We might make an educated guess $\hat{\sigma}_j$ and then employ $n_j \propto \omega_j\hat{\sigma}_j$. The optimal allocation only depends on $\sigma_1,\dots,\sigma_J$ through ratios σ_j/σ_k for $j \neq k$, and so only the ratios $\hat{\sigma}_j/\hat{\sigma}_k$ need to be accurate. Non-proportional allocations carry some risk. The optimal allocation assuming $\sigma_j = \hat{\sigma}_j$ can be worse than proportional allocation if it should turn out that σ_j are not proportional to $\hat{\sigma}_j$. It can even give higher variance than ordinary Monte Carlo sampling, completely defeating the effort put into stratification.

From results in survey sampling (Cochran, 1977), it is known how to construct theoretically optimal strata. The variance minimizing strata take the form $\mathcal{D}_k = \{\boldsymbol{x} \mid a_{k-1} \leqslant f(\boldsymbol{x}) < a_k\}$ for some constants $a_0 < a_1 < \cdots < a_J$. There are also guidelines for choosing the a_j. In practice we cannot usually locate the contours of f and even when we can it will usually be very hard to sample between them. But the intuition is still valuable: we want strata within which f is as flat as possible.

8.5 Example: stratified compound Poisson

Compound Poisson models are commonly used for rainfall. Here we will look at stratifying such a model.

In our model setting, the number of rainfall events (storms) in the coming month is $S \sim \mathrm{Poi}(\lambda)$ with $\lambda = 2.9$. The depth of rainfall in storm i is $D_i \sim \mathrm{Weib}(k,\sigma)$ with shape $k = 0.8$ and scale $\sigma = 3$ (centimeters) and the storms are independent. If the total rainfall is below 5 centimeters then an emergency water allocation will be imposed.

The total rainfall is thus $X = \sum_{s=1}^{S} D_s$ taking the value 0 when $S = 0$. It is easy to get the mean and variance of X, but here we want $\mathbb{P}(\boldsymbol{X} < 5)$, that is $\mathbb{E}(f(\boldsymbol{X}))$ where $f(\boldsymbol{X}) = \mathbb{1}_{\boldsymbol{X}<5}$. In a direct simulation, depicted in Figure 8.3,

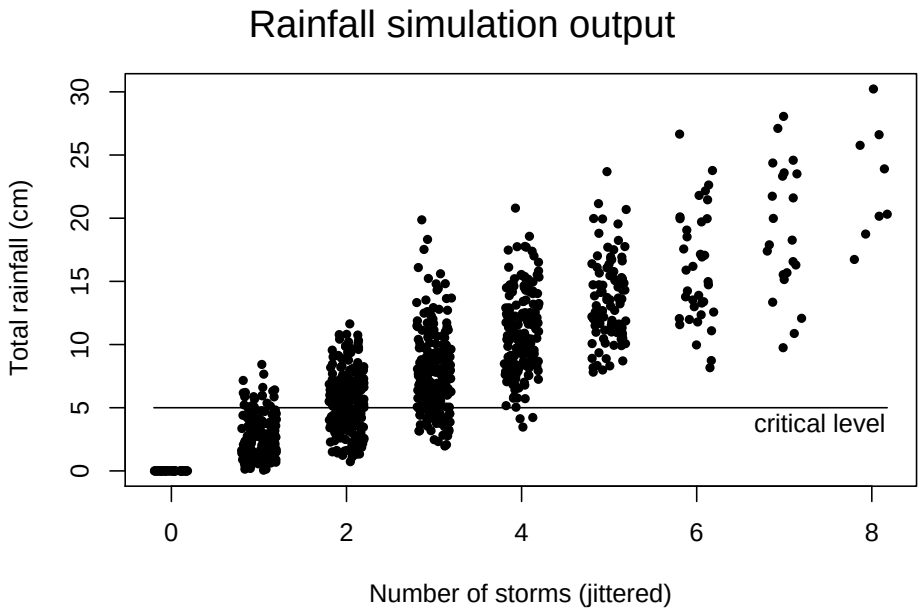

Figure 8.3: This figure depicts 1000 simulations of the compound Poisson model for rainfall described in the text.

the rainfall was below the critical level 353 times out of 1000. Thus the estimate of $\mathbb{P}(\boldsymbol{X} < 5)$ is $\hat{\mu} = 0.353$. Because this probability is not near 0 or 1 a simple 99% confidence interval of $\hat{\mu} \pm 2.58\sqrt{\hat{\mu}(1-\hat{\mu})/n}$ is adequate, and it yields the confidence interval $0.314 \leqslant \mathbb{P}(\boldsymbol{X} < 5) \leqslant 0.392$.

From simple Monte Carlo, we learn that the probability of a critically low total rainfall is roughly 30 to 40 percent. From Figure 8.3 we see that this probability depends strongly on the number of rainfall events.

Consider stratifying S according to a proportional allocation. The number of times $S = s$ in 1000 trials should be $n_s = 1000\, e^{-\lambda}\lambda^s/s!$ where $\lambda = 2.9$. Two issues come up immediately. First, the sample sizes n_s are not integers. That is not a serious problem. We can use rounded sample sizes, in an approximately proportional allocation and still obtain an unbiased estimate of $\mathbb{P}(\boldsymbol{X} < 5)$ and a workable variance estimate. The second issue to come up is that when $S = 0$ we don't really need to simulate at all. In that case we are sure that $\boldsymbol{X} < 5$. For the second issue we will take $n_0 = 2$. That way we can use the plain stratified sampling formulas (8.6) and (8.10), and we only waste 2 of 1000 simulations on the foregone conclusion that with no storms there will be a water shortage.

Taking $n_0 = 2$ samples with $S = 0$ and allocating the remaining 998 in proportion to $\mathbb{P}(S = s)/(1 - \mathbb{P}(S = 0))$ we get the counts

s	0	1	2	3	4	5	$\geqslant 6$
n_s	2	169	244	236	171	99	79

s	ω_s	n_s	T_s	$\hat{\mu}_s$	$\hat{\sigma}_s^2$
0	0.055	2	2	1.000	0.000
1	0.160	169	152	0.899	0.091
2	0.231	244	111	0.455	0.249
3	0.224	236	33	0.140	0.121
4	0.162	171	3	0.018	0.017
5	0.094	99	1	0.010	0.010
6+	0.074	79	1	0.013	0.013

Table 8.2: This table shows the results of a stratified simulation of the compound Poisson rainfall model from the text. Here s is the number of storms. The last stratum is for $s \geqslant 6$. Continuing, ω_s is $\mathbb{P}(S = s)$ under a Poisson model, and n_s is the number of simulations allocated to $S = s$. Of n_s trials, there were T_s below the critical level. Then $\hat{\mu}_s$ and $\hat{\sigma}_s^2$ are estimated within stratum means and variances.

where the values from 6 on up have been merged into one stratum.

The $S \geqslant 6$ stratum is more complicated to sample from than the others. One way is to first find $q_6 = \sum_{s=0}^{6} e^{-\lambda}\lambda^s/s!$. Then draw $S = F_\lambda^{-1}(q_6 + (1-q_6)U)$ where $U \sim \mathbf{U}(0,1)$ and F_λ is the Poi(λ) CDF.

The results of this simulation are shown in Table 8.2. Using those values, the estimated probability of a shortage is $\hat{\mu}_{\text{strat}} = \sum_s \omega_s\hat{\mu}_s \doteq 0.334$. Using equation (8.10), $\widehat{\text{Var}}(\hat{\mu}_{\text{strat}}) = \sum_s \omega_s^2\hat{\sigma}_s^2/n_s \doteq 9.84 \times 10^{-5}$. The plain Monte Carlo simulation has an estimated variance of $\hat{p}(1-\hat{p})/n \doteq 0.353\times 0.643/1000 \doteq 2.28 \times 10^{-4}$, about 2.3 times as large as the estimated variance for stratified sampling.

This value 2.3 is only an estimate, but it turns out to be close to correct. In 10,000 independent replications of both methods the sample variance of the 10,000 plain Monte Carlo simulation answers was 2.24 times as large as that of the 10,000 stratified sampling answers.

A variance reduction of just over 2–fold is helpful but not enormous. Such a variance reduction would only justify the extra complexity of stratified sampling, if we needed to run many simulations of this sort.

The estimated factor of 2.24 does not take into account running time. Stratification has the possibility of being slightly faster here because most of the samples are deterministic: instead of sampling 1000 Poisson random variables, we generate 79 variables from the right tail of the Poisson distribution and use pre-chosen values for the other 921 Poisson random variables.

A further modest variance reduction can be obtained by reducing the number of observations with $s \geqslant 5$, increasing the number with $s = 2$ or 3, and replacing the estimate from $s = 1$ by $\mathbb{P}(\text{Weib}(k,\sigma) \leqslant 5)$. None of these steps can bring a dramatic increase in accuracy because the strata $s = 2$ and 3 have high variance. Stratifying on S cannot help with the variance of $f(\boldsymbol{X})$ given $S = s$.

8.6 Common random numbers

Suppose that f and g are closely related functions and that we want to find $\mathbb{E}(f(\boldsymbol{X}) - g(\boldsymbol{X}))$ for $\boldsymbol{X} \sim p$. Perhaps $f(\boldsymbol{x}) = h(\boldsymbol{x}, \theta)$ for a parameter $\theta \in \mathbb{R}^p$, and then to study the effect of θ we look at $g(\boldsymbol{x}) = h(\boldsymbol{x}, \widetilde{\theta})$ for some $\widetilde{\theta} \neq \theta$. We assume at first that neither f nor g (nor h) makes any use of random numbers other than $\boldsymbol{X}$. Later we relax that assumption.

Because $\mathbb{E}(f(\boldsymbol{X})-g(\boldsymbol{X})) = \mathbb{E}(f(\boldsymbol{X}))-\mathbb{E}(g(\boldsymbol{X}))$ we clearly have two different ways to go. We could estimate the difference by

$$\widehat{D}_{\mathrm{com}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i) - g(\boldsymbol{X}_i), \tag{8.19}$$

for $\boldsymbol{X}_i \overset{\mathrm{iid}}{\sim} p$, or by differencing averages

$$\widehat{D}_{\mathrm{ind}} = \frac{1}{n_1}\sum_{i=1}^{n_1} f(\boldsymbol{X}_{i1}) - \frac{1}{n_2}\sum_{i=1}^{n_2} g(\boldsymbol{X}_{i2}) \tag{8.20}$$

for $\boldsymbol{X}_{ij} \overset{\mathrm{iid}}{\sim} p$. Taking $n = n_1 = n_2$ makes the computing costs in (8.19) and (8.20) comparable, assuming that costs of computing f and g dominate those of generating $\boldsymbol{X}$.

The sampling variances of these methods are

$$\begin{aligned}\mathrm{Var}(\widehat{D}_{\mathrm{com}}) &= \frac{1}{n}\bigl(\sigma_f^2 + \sigma_g^2 - 2\rho\sigma_f\sigma_g\bigr)\\ \mathrm{Var}(\widehat{D}_{\mathrm{ind}}) &= \frac{1}{n}\bigl(\sigma_f^2 + \sigma_g^2\bigr),\end{aligned} \tag{8.21}$$

where σ_f^2 and σ_g^2 are individual function variances and $\rho = \mathrm{Corr}(f(\boldsymbol{X}), g(\boldsymbol{X}))$. When $\rho > 0$ we are better off using common random numbers. There is no guarantee that $\rho > 0$. When f and g compute similar quantities then we anticipate that $\rho > 0$, and if so, then $\widehat{D}_{\mathrm{com}}$ is more effective than $\widehat{D}_{\mathrm{ind}}$.

Most people would instinctively use the common variates. So at first sight, the method looks more like avoiding a variance increase than engineering a variance decrease. Later, when we relax the rule forbidding f, g, and h to use other sources of randomness, we will find that retaining some common random numbers requires considerable care in synchronization. The added complexity might well tip the balance against using common random numbers.

Much the same problem arises if we are comparing $\mathbb{E}(f(\boldsymbol{X}))$ for $\boldsymbol{X} \sim p$ and $\mathbb{E}(f(\widetilde{\boldsymbol{X}}))$ for $\widetilde{\boldsymbol{X}} \sim \widetilde{p}$. Sometimes we can rewrite that problem in terms of common random variables that get transformed to a different distribution before f is applied. For instance, if the first simulation has $X_i \overset{\mathrm{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ and the second has $\widetilde{X}_i \overset{\mathrm{iid}}{\sim} \mathcal{N}(\widetilde{\mu}, \widetilde{\sigma}^2)$ then we can sample $Z_i \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,1)$ and use

$$\widehat{D}_{\mathrm{com}} = \frac{1}{n}\sum_{i=1}^{n} f(\mu + \sigma Z_i) - f(\widetilde{\mu} + \widetilde{\sigma} Z_i).$$

More generally, when $\boldsymbol{X}_i$ is generated via a transformation $\Psi(\boldsymbol{U}_i;\theta)$ of $\boldsymbol{U}_i \sim \mathbf{U}(0,1)^s$ then we can average $f(\Psi(\boldsymbol{U}_i;\theta)) - f(\Psi(\boldsymbol{U}_i;\widetilde{\theta}))$.

Acceptance-rejection sampling of $\boldsymbol{X}$ does not fit cleanly into this framework, because the number s of needed uniform random variables is not fixed and may vary with θ.

The construction above is a ***coupling*** of the random vectors $\boldsymbol{X}$ and $\widetilde{\boldsymbol{X}}$. Any joint distribution on $(\boldsymbol{X}, \widetilde{\boldsymbol{X}})$ with $\boldsymbol{X} \sim p$ and $\widetilde{\boldsymbol{X}} \sim \widetilde{p}$ is a coupling. Common random numbers provide a particularly close coupling between $\boldsymbol{X}$ and $\widetilde{\boldsymbol{X}}$.

Example: dosage content uniformity

Medicines are typically sold with a label claim giving the amount of active ingredient that should be in each dose. The actual amount fluctuates but should be close to the claim. Sampling schemes are used to determine whether a given lot has high enough quality. The average dose should be close to the target and the standard deviation should not be too large.

There are many different types of test, depending on the product (tablet, capsule, aerosol, skin patch, etc.). Here is one, based on the US Pharmacopeial Convention content uniformity test. We will measure the dose as a percentage of the label claim, and assume that the target value is 100% of label claim. In some instances targets over 100% are considered, perhaps to compensate for declining dosage in storage.

To describe the test, we need to introduce the function

$$M(x) = \begin{cases} 98.5, & x < 98.5 \\ x, & 98.5 \leqslant x \leqslant 101.5 \\ 101.5, & x > 101.5. \end{cases} \tag{8.22}$$

This function will be used to make the test less sensitive to tiny fluctuations in the average dose. Exercise 8.22 looks at whether using $M(x)$ makes any difference to the acceptance probability.

The test first samples 10 units, getting measured values $x_1, \dots, x_{10}$. Then the values

$$\bar{x}_1 = \frac{1}{10}\sum_{j=1}^{10} x_j, \quad s_1^2 = \frac{1}{9}\sum_{j=1}^{10}(x_j - \bar{x}_1)^2, \quad \text{and} \quad M_1 = M(\bar{x}_1)$$

are computed. The lot passes if $|\bar{x}_1 - M_1| + 2.4 s_1 \leqslant 15$. Otherwise, 20 more units are sampled giving $x_{11}, \dots, x_{30}$. Then the values

$$\bar{x}_2 = \frac{1}{30}\sum_{j=1}^{30} x_j, \quad s_2^2 = \frac{1}{29}\sum_{j=1}^{30}(x_j - \bar{x}_2)^2, \quad \text{and} \quad M_2 = M(\bar{x}_2)$$

are computed. The lot passes if $|\bar{x}_2 - M_2| + 2.0 s_2 \leqslant 15$ and $\min_{1\leqslant j\leqslant 30} x_j \geqslant 0.75 M_2$ and $\max_{1\leqslant j\leqslant 30} x_j \leqslant 1.25 M_2$. Otherwise it fails.

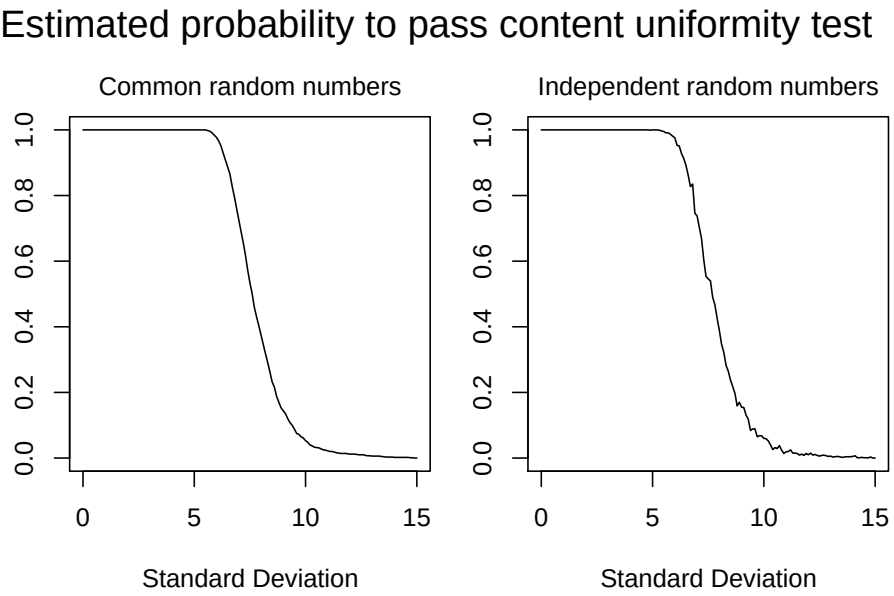

Figure 8.4: Each panel shows the estimated probability of passing the content uniformity test for $X_i \sim \mathcal{N}(100, \sigma^2)$ as the standard deviation increases from 0 to 15 units. The smooth curve on the left is based on common random numbers. The rougher curve on the right uses independent random numbers. Both were based on $n = 1000$ replications.

When the quality is high, the product usually passes at the first stage, and then the two stage test saves time and expense. But the two stage test is not amenable to closed form analysis even when $x_j \sim \mathcal{N}(\mu, \sigma^2)$. Monte Carlo methods are well suited to studying the probability of passing the test.

A direct simulation of the process is easy to do. But suppose that we want to compare the effects of varying μ and σ on the passage probability. Then it makes sense to use a common random number scheme with $Z_1, \dots, Z_{30}$ sampled independently from $\mathcal{N}(0, 1)$ and $X_j = \mu + \sigma Z_j$ for $j = 1, \dots, 30$. To keep the simulation synchronized, we always reserve the values $Z_{11}, \dots, Z_{30}$ for the second stage, even when the test is accepted at stage 1.

When $\mu = 100$, the test will tend to fail if σ is high enough. Figure 8.4 shows Monte Carlo estimates of the probability of passing the uniformity test for $X_j \sim \mathcal{N}(100, \sigma^2)$ with $0 \leqslant \sigma \leqslant 15$. The probability of passing is very high for $\sigma \leqslant 5.5$ or so, but then it starts to drop quickly. When common random numbers are used, the estimated probability is very smooth, and also monotone, in σ. When independent random numbers are used, the estimated probability is non-monotone and the non-smoothness is even visible to the eye.

For large enough n, the non-smoothness would not be visible, but it would still result in less accurate estimation of differences in acceptance probability.

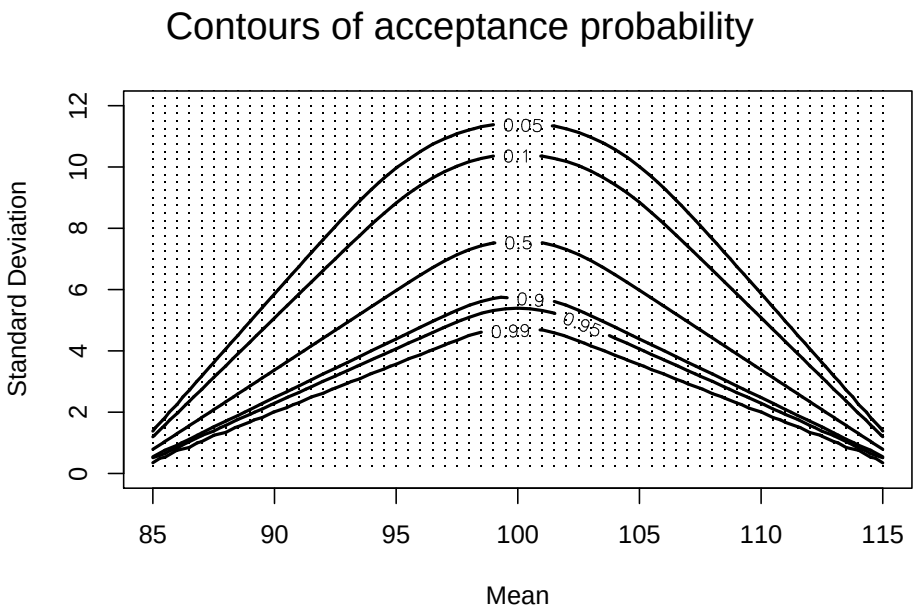

Figure 8.5: This plot shows contours of the acceptance probability of the content uniformity test when the data are $\mathcal{N}(\mu, \sigma^2)$. The horizontal axis is μ and the vertical axis is σ. Monte Carlo sampling with $n = 100{,}000$ points was run at each point of the grid shown in light dots. Values of μ run from 85 to 115 in steps of 0.5, while σ runs from 0.25 to 12.0 in steps of 0.25. Common random numbers were used.

The acceptance probability is mapped out as a function of μ and σ in Figure 8.5. That figure was created by using a common random numbers Monte Carlo sample on a grid of (μ, σ) pairs. There is a roughly triangular region in the (μ, σ) plane where the success probability is over 99%. Because the probability is between 99 and 100 percent there, and is monotone in σ and $|\mu - 100|$, the surface is very flat within this triangle. The region with 99.9% success probability (not shown) is just barely smaller than the one with 99% probability. There is a tiny bit of wiggle in some of the contours partly because the grid spacing is wide and partly because those contours go through a region where failures are rare events.

Implementing common random numbers

We want to estimate $\mu_j = \mathbb{E}(h(\boldsymbol{X}, \theta_j))$ for $j = 1, \ldots, m$ using n random inputs $\boldsymbol{X}_i$, for $i = 1, \ldots, n$. The content uniformity example had a large value of m but in the simplest case, $m = 2$ and we're interested in $\mu_1 - \mu_2$. We still assume that h really is a function of $\boldsymbol{X}$ and θ and in particular our implementation of h does not cause more random numbers to be generated.

Algorithm 8.1 Common random numbers algorithm I

`setseed`(seed)
$\hat{\mu}_j \leftarrow 0, \quad 1 \leqslant j \leqslant m$
for $i = 1$ to n **do**
  $\boldsymbol{X}_i \sim p$
  $\hat{\mu}_j \leftarrow \hat{\mu}_j + h(\boldsymbol{X}_i, \theta_j), \quad 1 \leqslant j \leqslant m$
$\hat{\mu}_j \leftarrow \hat{\mu}_j/n, \quad 1 \leqslant j \leqslant m$
deliver $\hat{\mu}_1, \ldots, \hat{\mu}_m$

This algorithm shows the method of common random numbers with the outer loop over random samples. The only random numbers used in h are from $\boldsymbol{X}_i$. Setting the seed keeps the $\boldsymbol{X}_i$ reproducible if we change our list of θ_j. The vectorized approach of equation (8.23) may be convenient.

We can run a nested loop over samples indexed by i and parameter values indexed by j. There are two main approaches that we can take, depending on which is the outer loop.

Algorithm 8.1 shows common random numbers with the outer loop over $\boldsymbol{X}_i$ for $i = 1, \ldots, n$. When $\boldsymbol{X}_i$ is multi-dimensional we have to make sure that every component of $\boldsymbol{X}$ needed for any value of θ_j is provided. In the content uniformity problem (page 18) we generate Z_{11} for every simulated batch even though some only use $Z_1, \ldots, Z_{10}$.

A vectorized implementation of Algorithm 8.1 is advantageous. It uses a function H that takes $\boldsymbol{X}$ and a list $\Theta = (\theta_1, \ldots, \theta_m)$ of parameter values. This H returns a list $(h(\boldsymbol{X}, \theta_1), \ldots, h(\boldsymbol{X}, \theta_m))$ and the simulation computes

$$(\hat{\mu}_1, \ldots, \hat{\mu}_m) = \frac{1}{n} \sum_{i=1}^{n} H(\boldsymbol{X}_i, \Theta). \tag{8.23}$$

This vectorized H makes it easier to separate the code that creates Θ from that which evaluates h.

Algorithm 8.2 shows common random numbers with the outer loop over the parameters. It regenerates all n vectors $\boldsymbol{X}_i$ for each j. To keep these vectors synchronized it keeps resetting the random seed. If we look at the output from Algorithm 8.1 partway through the computation, we will see incomplete estimates for all of the θ_j. If we do that for Algorithm 8.2 we will see completed estimates for a subset of the θ_j.

Now suppose that we relax our constraint on h and allow it to sample random numbers. That creates some messy synchronization issues described on page 36 of the end notes. Algorithm 8.1 is more robust to this change than Algorithm 8.2, but both could bring unpleasant surprises. Such a relaxation leaves us with only partially common numbers that we look at next.

Algorithm 8.2 Common random numbers algorithm II

for $j = 1$ to m **do**
  `setseed`(seed), $\hat{\mu}_j \leftarrow 0$
  for $i = 1$ to n **do**
    $\boldsymbol{X}_i \sim p$
    $\hat{\mu}_j \leftarrow \hat{\mu}_j + h(\boldsymbol{X}_i, \theta_j)$
  $\hat{\mu}_j \leftarrow \hat{\mu}_j / n$
deliver $\hat{\mu}_1, \ldots, \hat{\mu}_m$

This algorithm shows the method of common random numbers with the outer loop over the parameter list. It keeps resetting the seed and regenerating the data. The only random numbers used in h are from $\boldsymbol{X}_i$.

Partial common random numbers

Sometimes we can take some but not all of the random variables in two simulations to be common. For instance, suppose that we want to simulate how a coffee shop operates. There is a process by which customers arrive and choose what to order. Then another process defines how quickly their order is fulfilled. We might want to compare two or more service processes. Perhaps the shop adds one more barista at peak hours, or changes how the customers line up, or buys new equipment. Under any of these changes we should be able to run the same sequence of simulated customers through the shop. But there may be no practical way to implement any form of common service times.

In general, we may be trying to find $\mu = \mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y}) - g(\boldsymbol{X}, \boldsymbol{Z}))$ for independent inputs $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$. In the coffee shop example, $\boldsymbol{X}$ drives the customer arrivals while $\boldsymbol{Y}$ (or $\boldsymbol{Z}$) determines their service times conditionally on the set of arrival times. We can use

$$\hat{\mu}_{\mathrm{ind}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i, \boldsymbol{Y}_i) - g(\widetilde{\boldsymbol{X}}_i, \boldsymbol{Z}_i)$$

where $\boldsymbol{X}_i$, $\widetilde{\boldsymbol{X}}_i$, $\boldsymbol{Y}_i$ and $\boldsymbol{Z}_i$ are mutually independent. To make a more accurate comparison we would rather have $\widetilde{\boldsymbol{X}}_i = \boldsymbol{X}_i$. Then we use

$$\hat{\mu}_{\mathrm{pcom}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i, \boldsymbol{Y}_i) - g(\boldsymbol{X}_i, \boldsymbol{Z}_i)$$

This is only a 'partial common random numbers' algorithm because some but not all of the inputs are common.

Example 8.1 (Coupling Poisson variables and processes)**.** Suppose that $X \sim \mathrm{Poi}(\mu)$ and $Y \sim \mathrm{Poi}(\eta)$ with $0 < \mu < \eta$. We can sample X and Y by inversion from a common random variable $U \sim \mathbf{U}(0,1)$ and they will be closely coupled. We can also simulate $X \sim \mathrm{Poi}(\mu)$, $Z \sim \mathrm{Poi}(\eta - \mu)$, and take $Y = X + Z$. This second approach does not generate quite as close a connection between X and Y but it underlies a useful generalization to Poisson processes.

Let $\lambda_j \geqslant 0$ for $j = 1, 2$ be two intensity functions on $[0, T]$ with corresponding cumulative intensity functions $\Lambda_j(t) = \int_0^t \lambda_j(t)\,\mathrm{d}t$. We can sample these two processes via $T_{i,j} = \Lambda_j^{-1}(\Lambda_j(T_{i-1,j}) + E_i)$, $j = 1, 2$, using the common random numbers $E_i \overset{\text{iid}}{\sim} \text{Exp}(1)$.

The processes $T_{i,1}$ and $T_{i,2}$ are simulated from common random numbers but they won't have any common event times. When common event times are desired, we can proceed as follows. We define $\underline{\lambda}(t) = \min(\lambda_1(t), \lambda_2(t))$ and $\lambda_j^*(t) = \lambda_j(t) - \underline{\lambda}(t)$ for $j = 1, 2$. These have cumulative intensities $\underline{\Lambda}$ and Λ_j^*, respectively, and they generate Poisson process realizations $\underline{T}_i$ for $i = 1, \ldots, \underline{N}$ and T_j^* for $i = 1, \ldots, N_j^*$. Now we take

$$\{T_{1,1}, \ldots, T_{N_1,1}\} = \{\underline{T}_1, \ldots, \underline{T}_{\underline{N}}\} \cup \{T_{1,1}^*, \ldots, T_{N_1^*,1}^*\}, \quad \text{and}$$
$$\{T_{1,2}, \ldots, T_{N_2,2}\} = \{\underline{T}_1, \ldots, \underline{T}_{\underline{N}}\} \cup \{T_{1,2}^*, \ldots, T_{N_2^*,2}^*\}.$$

If necessary, we sort the points of each process. These processes share $\underline{N}$ common event times while having N_j^* unshared event times each. Anderson and Higham (2012) use this method to couple multilevel simulations of continuous time Markov chains.

Derivative estimation

An extreme instance of the value of common random numbers arises in estimating a derivative. Suppose that $\mu(\theta) = \mathbb{E}(h(\boldsymbol{X}, \theta))$ and that we want to estimate $\mu'(\theta_0) = \mathrm{d}\mu/\mathrm{d}\theta|_{\theta=\theta_0}$ We assume that $h(\boldsymbol{x}, \theta)$ is well behaved enough to satisfy

$$\frac{\mathrm{d}}{\mathrm{d}\theta} \int h(\boldsymbol{x}, \theta) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int \frac{\partial}{\partial\theta} h(\boldsymbol{x}, \theta) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

at $\theta = \theta_0$. If we can compute the needed partial derivative, then we can take

$$\hat{\mu}'(\theta_0) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial\theta} h(\boldsymbol{X}_i, \theta)$$

for $\boldsymbol{X}_i \sim p$. Otherwise, we may need to use divided differences, such as the forward or centered estimators,

$$\hat{\mu}'_F(\theta_0) = \frac{1}{n} \sum_{i=1}^{n} \frac{h(\boldsymbol{X}_i, \theta_0 + \epsilon) - h(\boldsymbol{X}_i, \theta_0)}{\epsilon}, \quad \text{or}$$
$$\hat{\mu}'_C(\theta_0) = \frac{1}{n} \sum_{i=1}^{n} \frac{h(\boldsymbol{X}_i, \theta_0 + \epsilon) - h(\boldsymbol{X}_i, \theta_0 - \epsilon)}{2\epsilon},$$

respectively, for some small $\epsilon > 0$.

Using common random variables we can take a very small $\epsilon > 0$, limited only by numerical stability of the required differences. By contrast, with independent random variables, the variance would be

$$\frac{\text{Var}(h(\boldsymbol{X}, \theta_0)) + \text{Var}(h(\boldsymbol{X}, \theta_0 + \epsilon))}{n\epsilon^2}$$

leading to certain failure as $\epsilon \to 0$.

If we cannot use common random numbers then there is a bias-variance tradeoff in choosing the optimal ϵ given the sample size n. We can sketch the result using Taylor series centered at θ_0 for each of $\theta_0 + \epsilon$ and $\theta_0 - \epsilon$. If h has three partial derivatives with respect to θ then

$$h(\boldsymbol{X}, \theta_0 \pm \epsilon) = h(\boldsymbol{X}, \theta_0) \pm \epsilon \frac{\partial}{\partial \theta} h(\boldsymbol{X}, \theta_0) + \frac{\epsilon^2}{2} \frac{\partial^2}{\partial \theta^2} h(\boldsymbol{X}, \theta_0) \pm \frac{\epsilon^3}{6} \frac{\partial^3}{\partial \theta^3} h(\boldsymbol{X}, \theta_\pm)$$

where $\theta_\pm$ is between θ_0 and $\theta_0 \pm \epsilon$ and may depend on $\boldsymbol{X}$. Therefore

$$\frac{h(\boldsymbol{X}, \theta_0 + \epsilon) - h(\boldsymbol{X}, \theta_0 - \epsilon)}{2\epsilon} = \frac{\partial}{\partial \theta} h(\boldsymbol{X}, \theta_0) + O_p(\epsilon^2).$$

The result is that the bias in $\hat{\mu}'_C(\theta_0)$ is $O_p(\epsilon^2)$ while the variance is $O(1/(n\epsilon^2))$. The optimal tradeoff has $\epsilon \propto n^{-1/6}$ with a mean squared error of $O(n^{-1/3})$. Some references on page 36 of the end notes give more information on estimating derivatives.

8.7 Conditioning

Sometimes we can do part of the problem in closed form, and then do the rest of it by Monte Carlo or some other numerical method. Suppose for example that we want to find $\mu = \int_0^1 \int_0^1 f(x, y)\, \mathrm{d}x\mathrm{d}y$ where $f(x, y) = e^{g(x)y}$. It is easy to integrate out y for fixed x, yielding $h(x) = (e^{g(x)} - 1)/g(x)$. Then we have a one dimensional problem, which may be simpler to handle. If g is complicated, such as $g(x) = \sqrt{5/4 + \cos(2\pi x)}$, then we cannot easily integrate x out of $h(x)$. Nor, it seems, can we integrate $f(x, y)$ over x for fixed y in closed form.

In general, suppose that $\boldsymbol{X} \in \mathbb{R}^k$ and $\boldsymbol{Y} \in \mathbb{R}^{d-k}$ are random vectors and that we want to estimate $\mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y}))$. The natural estimate is $\hat{\mu} = (1/n)\sum_{i=1}^n f(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ where $(\boldsymbol{X}_i, \boldsymbol{Y}_i) \in \mathbb{R}^d$ are independent samples from the joint distribution of $(\boldsymbol{X}, \boldsymbol{Y})$. Now let $h(\boldsymbol{x}) = \mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y}) \mid \boldsymbol{X} = \boldsymbol{x})$. We might also estimate μ by

$$\hat{\mu}_{\mathrm{cond}} = \frac{1}{n} \sum_{i=1}^n h(\boldsymbol{X}_i) \tag{8.24}$$

where $\boldsymbol{X}_i$ are independently sampled from the distribution of $\boldsymbol{X}$. The justification for the method is that $\mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y})) = \mathbb{E}(\mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y}) \mid \boldsymbol{X})) = \mathbb{E}(h(\boldsymbol{X}))$. The function $h(\cdot)$ gives the conditional mean of $\boldsymbol{Y}$ in closed form and then we complete the job by Monte Carlo sampling. The method is called ***conditioning***, or ***conditional Monte Carlo***, for obvious reasons. The main requirement for conditioning is that we must be able to compute $h(\cdot)$. We also need a method for sampling $\boldsymbol{X}$, but we have that already if we can sample $(\boldsymbol{X}, \boldsymbol{Y})$ jointly.

We easily find that

$$\mathrm{Var}(\hat{\mu}_{\mathrm{cond}}) = \frac{1}{n}\mathrm{Var}(h(\boldsymbol{X})) = \frac{1}{n}\mathrm{Var}(\mathbb{E}(f(\boldsymbol{X}, \boldsymbol{Y}) \mid \boldsymbol{X})).$$

Recalling the elementary expression

$$\mathrm{Var}(f(\boldsymbol{X},\boldsymbol{Y})) = \mathbb{E}(\mathrm{Var}(f(\boldsymbol{X},\boldsymbol{Y}) \mid \boldsymbol{X})) + \mathrm{Var}(\mathbb{E}(f(\boldsymbol{X},\boldsymbol{Y}) \mid \boldsymbol{X}))$$

it is immediately clear that conditional Monte Carlo cannot have higher variance than ordinary Monte Carlo sampling of f has and will typically have strictly smaller variance. We summarize that finding as follows:

Theorem 8.1. *Let $(\boldsymbol{X},\boldsymbol{Y})$ have joint distribution F and let $f(\boldsymbol{x},\boldsymbol{y})$ satisfy $\mathrm{Var}(f(\boldsymbol{X},\boldsymbol{Y})) = \sigma^2 < \infty$. Define $h(\boldsymbol{x}) = \mathbb{E}(f(\boldsymbol{X},\boldsymbol{Y}) \mid \boldsymbol{X} = \boldsymbol{x})$ for $(\boldsymbol{X},\boldsymbol{Y}) \sim F$. Suppose that $(\boldsymbol{X}_i,\boldsymbol{Y}_i) \sim F$. Then*

$$\mathrm{Var}\biggl(\frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{X}_i)\biggr) \leqslant \mathrm{Var}\biggl(\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i,\boldsymbol{Y}_i)\biggr).$$

Conditioning is a special case of ***derandomization***. The function $f(\boldsymbol{X},\boldsymbol{Y})$ has two sources of randomness, $\boldsymbol{X}$ and $\boldsymbol{Y}$. For any given $\boldsymbol{x}$ and random $\boldsymbol{Y}$ we replace the random value $f(\boldsymbol{x},\boldsymbol{Y})$ by its expectation $h(\boldsymbol{x})$, removing one of the two sources of randomness. For the function $f(x,y) = e^{g(x)y}$ at the beginning of this section, derandomization brings a nice, but not overwhelming, variance reduction. See Exercise 8.9.

Conditioning is sometimes called ***Rao-Blackwellization*** in reference to the Rao-Blackwell theorem in theoretical statistics. In that theorem, the quantity being conditioned on has to obey quite stringent conditions. Those conditions usually don't hold in Monte Carlo applications and, from Theorem 8.1, we don't need them. As a result, the term Rao-Blackwellization is not really descriptive of the way conditioning is used in Monte Carlo sampling.

Even though derandomization by conditioning always reduces variance, it is not always worth doing. We could find our estimate is less efficient if computing h costs much more than computing f does. For instance, to average

$$f(\boldsymbol{x}) = \cos\Bigl(g(x_1) + \sum_{j=1}^{d} a_j x_j\Bigr)$$

over $\boldsymbol{x} \in (0,1)^d$, we can derandomize and average

$$\frac{1}{a_d}\biggl(\sin\Bigl(g(x_1) + \sum_{j=1}^{d-1} a_j x_j + a_d\Bigr) - \sin\Bigl(g(x_1) + \sum_{j=1}^{d-1} a_j x_j\Bigr)\biggr)$$

over $(0,1)^{d-1}$ instead. We have reduced the variance but will have nearly doubled the cost, if evaluating $\sin(\cdot)$ is the most expensive part of computing f. Derandomizing $d-1$ times would leave us with a one dimensional integrand that requires 2^{d-1} sinusoids to evaluate.

Example 8.2 (Hit or miss)**.** Let $C = \{\,(x,y) \mid a \leqslant x \leqslant b,\ 0 \leqslant y \leqslant f(x)\,\}$. Suppose that $f(x) \leqslant c$ holds for $a \leqslant x \leqslant b$. Then the hit or miss Monte Carlo

estimate of $\mathbf{vol}(C)$ is

$$\widehat{\mathbf{vol}}(C) = \frac{c(b-a)}{n}\sum_{i=1}^{n} \mathbb{1}_{Y_i \leqslant f(X_i)}$$

where $(X_i, Y_i) \sim \mathbf{U}([a,b]\times[0,c])$ are independent for $i = 1,\dots,n$. Now $h(x) = \mathbb{E}(\mathbb{1}_{Y\leqslant f(X)} \mid X = x) = f(x)/c$. Derandomizing hit or miss Monte Carlo by conditioning, yields the estimate

$$\frac{c(b-a)}{n}\sum_{i=1}^{n}\frac{f(X_i)}{c} = \frac{b-a}{n}\sum_{i=1}^{n} f(X_i).$$

The result is perhaps the most obvious way to estimate $\mathbf{vol}(C)$ by Monte Carlo, and it has lower variance than hit or miss. A case could be made for hit or miss when the average cost of determining whether $Y \leqslant f(X)$ holds is quite small compared to the cost of precisely evaluating $f(X)$ itself. But outside of such special circumstances, there is little reason to use hit or miss MC for finding the area under a curve.

Conditioning can be used in combination with other variance reduction methods. The most straightforward way is to apply those other methods to the problem of estimating $\mathbb{E}(h(\boldsymbol{X}))$. The combination of conditioning with stratified and/or antithetic sampling of $\boldsymbol{X}$ is thus simple, provided that the distribution of $\boldsymbol{X}$ is amenable to stratification or has some natural symmetry that we can exploit in antithetic sampling.

Conditioning brings a dimension reduction in addition to the variance reduction, because the dimension k of $\boldsymbol{X}$ is smaller than the dimension d, of $(\boldsymbol{X}, \boldsymbol{Y})$. When k is very small, then stratification methods or even quadrature can be used to compound the gain from conditioning. The example in §8.8 has $d = 38$ and $k = 1$.

8.8 Example: maximum Dirichlet

The gambler Allan Wilson once tabulated the results of 79,800 plays at a roulette table. Those values are given in the column labeled 'Wheel 1' in Table 8.3. The wheel on that table had 38 slots, numbered 1 through 36 along with 0 and 00, which we'll denote by 37 and 38 respectively. The wheel seemed to be imperfect, either due to manufacture or maintenance. The number 19 came up more often than any other.

Suppose that the counts $\boldsymbol{C} = (C_1,\dots,C_{38})$ for wheel 1 follow a $\mathrm{Mult}(N, \boldsymbol{p})$ distribution with $N = 79{,}800$ and $\boldsymbol{p} = (p_1,\dots,p_{38})$. If we adopt a prior distribution with $\boldsymbol{p} \sim \mathrm{Dir}(1,\dots,1)$ then the posterior distribution of $\boldsymbol{p}$ given that $\boldsymbol{C} = \boldsymbol{c}$ is $\mathrm{Dir}(\alpha_1,\dots,\alpha_{38})$ where $\alpha_j = c_j + 1$. For this posterior distribution, we would like to know $\mathbb{P}(p_{19} = \max_{1\leqslant j\leqslant 38} p_j)$, the probability that number 19 really does come up most often.

Number	Wheel 1	Wheel 2
00	2127	1288
1	2082	1234
13	2110	1261
36	2221	1251
24	2192	1164 w
3	2008	1438 b
15	2035	1264
34	2113	1335
22	2099	1342
5	2199	1232
17	2044	1326
32	2133	1302
20	1912 w	1227
7	1999	1192
11	1974	1278
30	2051	1336
26	1984	1296
9	2053	1298
28	2019	1205
0	2046	1189
2	1999	1171
14	2168	1279
35	2150	1315
23	2041	1296
4	2047	1256
16	2091	1304
33	2142	1304
21	2196	1351
6	2153	1281
18	2191	1392
31	2192	1306
19	2284 b	1330
8	2136	1266
12	2110	1224
29	2032	1190
25	2188	1229
10	2121	1320
27	2158	1336
Avg	2100	1279.16

Table 8.3: This table gives counts from two roulette wheels described in Wilson (1965, Appendix E). The best and worst holes, for the customer, are marked with b and w respectively.

In §5.4 we represented the Dirichlet distribution as normalized independent Gamma random variables. Here we can define $\boldsymbol{X} = (X_1, \dots, X_{38})$ where $X_j \sim \mathrm{Gam}(\alpha_j)$ are independent, and $p_j = X_j / \sum_{k=1}^{38} X_k$. Clearly p_{19} is the largest p_j if and only if X_{19} is the largest X_j. Therefore, we want to find $\mu = \mathbb{E}(f(\boldsymbol{X}))$ where

$$f(\boldsymbol{X}) = \begin{cases} 1, & X_{19} = \max_{1 \leqslant j \leqslant 38} X_j \\ 0, & X_{19} < \max_{1 \leqslant j \leqslant 38} X_j. \end{cases}$$

A direct Monte Carlo estimate of μ proceeds by repeatedly sampling $\boldsymbol{X} \in [0, \infty)^{38}$ and averaging $f(\boldsymbol{X})$. Here we condition on X_{19}. Given that $X_{19} = x_{19}$, the probability that X_{19} is largest is

$$h(x_{19}) = \prod_{j=1, j \neq 19}^{38} G_{\alpha_j}(x_{19}) \tag{8.25}$$

where $G_\alpha(x) = \int_0^x e^{-y} y^{\alpha-1} \, \mathrm{d}y / \Gamma(\alpha)$ is the CDF of the $\mathrm{Gam}(\alpha)$ distribution. To find the answer for this roulette wheel, do Exercise 8.10.

By conditioning, we replace $(1/n) \sum_{i=1}^n f(\boldsymbol{X}_i)$ where $X_{ij} \sim \mathrm{Gam}(\alpha_j)$ are independent by $(1/n) \sum_{i=1}^n h(Y_i)$ where $Y_i \sim \mathrm{Gam}(\alpha_{19})$ are independent.

Computations for the function $h(y)$ could, in some instances, underflow. That does not happen for the roulette example, but if we want to get the probability that the apparent worst number is actually the best, the values of h become very small. Similarly for problems with higher dimensional Dirichlet distributions and more unequal counts, underflow is more likely. Underflow can be mitigated by working with software that computes $\log(G_{\alpha_j})$ directly. To find the probability that component j_0 is the largest of J components, we can define $\widetilde{h}(y) = \sum_{j=1, j \neq j_0}^{J} \log(G_{\alpha_j}(y))$ find $h^* = \max_{1 \leqslant i \leqslant n} \widetilde{h}(y_i)$ for the sampled y_i values and report the answer as $\exp(h^*)$ times $(1/n) \sum_{i=1}^n \exp(\widetilde{h}_i - h^*)$.

8.9 Control variates

We saw in §8.7 on conditioning how to get a better estimate by doing part of the problem in closed form. Control variates provide another way to exploit closed form results. With control variates we use some other problem, quite similar to our given one, but for which an exact answer is known. The precise meaning of 'similar' depends on how we will use this other problem, and more than one method is given below. As for 'exact', we will mean it literally, but in practice it may just mean known with an error negligible compared to Monte Carlo errors.

Suppose first that we want to find $\mu = \mathbb{E}(f(\boldsymbol{X}))$ and that we know the value $\theta = \mathbb{E}(h(\boldsymbol{X}))$ where $h(\boldsymbol{x}) \approx f(\boldsymbol{x})$. Letting $\hat{\mu} = (1/n) \sum_{i=1}^n f(\boldsymbol{X}_i)$ and $\hat{\theta} = (1/n) \sum_{i=1}^n h(\boldsymbol{X}_i)$ we can estimate μ by the ***difference estimator***

$$\hat{\mu}_{\mathrm{diff}} = \frac{1}{n} \sum_{i=1}^n \bigl(f(\boldsymbol{X}_i) - h(\boldsymbol{X}_i) \bigr) + \theta = \hat{\mu} - \hat{\theta} + \theta. \tag{8.26}$$

The expected value of $\hat{\mu}_{\text{diff}}$ is μ because $\mathbb{E}(\hat{\theta}) = \theta$. The variance of $\hat{\mu}_{\text{diff}}$ is

$$\text{Var}(\hat{\mu}_{\text{diff}}) = \frac{1}{n}\text{Var}(f(\boldsymbol{X}) - h(\boldsymbol{X})).$$

So if h is similar to f in the sense that the difference $f(\boldsymbol{X}) - h(\boldsymbol{X})$ has smaller variance than $f(\boldsymbol{X})$ has, we will get reduced variance by using $\hat{\mu}_{\text{diff}}$.

In this setting $h(\boldsymbol{X})$, the random variable whose mean is known, is the ***control variate***. The difference estimator is not the only way to use a control variate. The ratio and product estimators

$$\hat{\mu}_{\text{ratio}} = \hat{\mu}\,\theta/\hat{\theta}, \quad \text{and} \tag{8.27}$$

$$\hat{\mu}_{\text{prod}} = \hat{\mu}\,\hat{\theta}/\theta \tag{8.28}$$

respectively, are also used. These estimators are undefined when $\theta = 0$, but otherwise they generally converge to μ as $n \to \infty$. See Exercise 8.18 for the product estimator. The ratio and product estimators are usually biased because $\mathbb{E}(\hat{\theta}/\hat{\mu}) \neq \theta/\mu$ and $\mathbb{E}(\hat{\theta}\hat{\mu}) \neq \theta\mu$ in general. It is possible to generalize the control variate method in very complicated ways. Maybe we could use $\hat{\mu}\cos(\hat{\theta} - \theta)$ or some more imaginative quantity. But we don't. By far the most common way of using a control variate is through the regression estimator, considered next.

For a value $\beta \in \mathbb{R}$, the ***regression estimator*** of μ is

$$\hat{\mu}_\beta = \frac{1}{n}\sum_{i=1}^{n}\big(f(\boldsymbol{X}_i) - \beta h(\boldsymbol{X}_i)\big) + \beta\theta = \hat{\mu} - \beta(\hat{\theta} - \theta). \tag{8.29}$$

Taking $\beta = 0$ yields the simple MC estimator $\hat{\mu}$ and $\beta = 1$ gives us the difference estimator. The regression estimator is unbiased: $\mathbb{E}(\hat{\mu}_\beta) = \mu$ for all β because $\mathbb{E}(\hat{\theta}) = \theta$.

The variance of the regression estimator is

$$\text{Var}(\hat{\mu}_\beta) = \frac{1}{n}\big(\text{Var}(f(\boldsymbol{X})) - 2\beta\text{Cov}(f(\boldsymbol{X}), h(\boldsymbol{X})) + \beta^2\text{Var}(h(\boldsymbol{X}))\big).$$

By differentiating, we find that the best value of β is

$$\beta_{\text{opt}} = \frac{\text{Cov}(f(\boldsymbol{X}), h(\boldsymbol{X}))}{\text{Var}(h(\boldsymbol{X}))} = \frac{\mathbb{E}\big((h(\boldsymbol{X}) - \theta)f(\boldsymbol{X})\big)}{\mathbb{E}\big((h(\boldsymbol{X}) - \theta)^2\big)},$$

and after some algebra, the resulting minimal variance is

$$\text{Var}(\hat{\mu}_{\beta_{\text{opt}}}) = \frac{\sigma^2}{n}\,(1 - \rho^2),$$

where $\rho = \text{Corr}(f(\boldsymbol{X}), h(\boldsymbol{X}))$. In the regression estimator, any control variate that correlates with f is helpful, even one that correlates negatively.

In practice we don't know β_{opt} and so we estimate it by

$$\hat{\beta} = \sum_{i=1}^{n}(f(\boldsymbol{X}_i) - \bar{f})(h(\boldsymbol{X}_i) - \bar{h})\Big/\sum_{i=1}^{n}(h(\boldsymbol{X}_i) - \bar{h})^2,$$

where $\bar{f} = (1/n)\sum_{i=1}^n f(\boldsymbol{X}_i)$ and $\bar{h} = (1/n)\sum_{i=1}^n h(\boldsymbol{X}_i)$. Then the regression estimator of μ is $\hat{\mu}_{\hat{\beta}}$. In general $\mathbb{E}(\hat{\mu}_{\hat{\beta}}) \neq \mu$, but this bias is usually small. We postpone study of the bias until later (equation (8.34)) when we consider multiple control variates. The estimated variance of $\hat{\mu}_{\hat{\beta}}$ is

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\hat{\beta}}) = \frac{1}{n^2}\sum_{i=1}^n \bigl(f(\boldsymbol{X}_i) - \hat{\mu}_{\hat{\beta}} - \hat{\beta}(h(\boldsymbol{X}_i) - \bar{h})\bigr)^2,$$

and a 99% confidence interval is $\hat{\mu}_{\hat{\beta}} \pm 2.58\sqrt{\widehat{\mathrm{Var}}(\hat{\mu}_{\hat{\beta}})}$.

The variance with a control variate is $\sigma^2(1-\rho^2)/n$ which is never worse than σ^2/n and usually better. Whether the control variate is helpful ultimately depends on how much it costs to use it. Suppose that the total cost of generating $\boldsymbol{X}_i$ and then computing $f(\boldsymbol{X}_i)$ is, on average, c_f. Let c_h be the extra cost incurred by the control variate on average. That includes the cost to evaluate $h(\boldsymbol{X}_i)$ but not the cost of sampling $\boldsymbol{X}_i$. We will suppose that the cost to compute $\hat{\beta}$ is small. If not then c_h should be increased to reflect it. Control variates improve efficiency when $(1-\rho^2)(c_f+c_h) < c_f$, that is when $|\rho| > \sqrt{c_h/(c_f+c_h)}$. For illustration, if $c_h = c_f$ then we need $|\rho| > \sqrt{1/2} \doteq 0.71$ in order to benefit from the control variate.

Example 8.3 (Arithmetic and geometric Asian option)**.** A well known and very effective control variate arises in finance. Let $f(\boldsymbol{X}) = \max(0, (1/m)\sum_{k=1}^m S(t_k) - K)$ be the value of an Asian call option, from §6.4, in terms of a geometric Brownian motion $S(t)$ generated from $\boldsymbol{X} \sim \mathbf{U}(0,1)^d$. Now let $h(\boldsymbol{X}) = \max(0, \prod_{k=1}^m S(t_k)^{1/m} - K)$, be the same option except that the arithmetic average has been replaced by a geometric average. The geometric average has a lognormal distribution. Thus θ can be computed by a one dimensional integral with respect to the normal probability density function. The result is the Black-Scholes formula.

A significant advantage of the regression estimator is that it generalizes easily to handle multiple control variates. The potential value is greatest when f is expensive but is approximately equal to a linear combination of inexpensive control variates.

Suppose that $\mathbb{E}(h_j(\boldsymbol{X})) = \theta_j$ are known values for $j = 1, \dots, J$. Let $h(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \dots, h_J(\boldsymbol{x}))^{\mathsf{T}}$ be a vector of functions with $\mathbb{E}(h(\boldsymbol{X})) = \theta = (\theta_1, \dots, \theta_J)^{\mathsf{T}}$, and let $\beta = (\beta_1, \dots, \beta_J)^{\mathsf{T}} \in \mathbb{R}^J$. The regression estimator for $J \geqslant 1$ is

$$\hat{\mu}_\beta = \frac{1}{n}\sum_{i=1}^n \bigl(f(\boldsymbol{X}_i) - \beta^{\mathsf{T}} h(\boldsymbol{X}_i)\bigr) + \beta^{\mathsf{T}}\theta = \hat{\mu} - \beta^{\mathsf{T}}\bar{H} + \beta^{\mathsf{T}}\theta \tag{8.30}$$

where $\bar{H} = (1/n)\sum_{i=1}^n h(\boldsymbol{X}_i)$. As before, $\mathbb{E}(\hat{\mu}_\beta) = \mu$.

The variance of $\hat{\mu}_\beta$ is σ_β^2/n where

$$\sigma_\beta^2 = \mathbb{E}\bigl(\bigl(f(\boldsymbol{X}) - \mu - \beta^{\mathsf{T}}(h(\boldsymbol{X}) - \theta)\bigr)^2\bigr). \tag{8.31}$$

Algorithm 8.3 Control variates by regression

given $f(\boldsymbol{x}_i)$, $h_j(\boldsymbol{x}_i)$, $\theta_j = \mathbb{E}(h_j(\boldsymbol{X}))$, $\quad i = 1, \dots, n, \quad j = 1, \dots, J$
$Y_i \leftarrow f(\boldsymbol{x}_i), \quad i = 1, \dots, n$
$Z_{ij} \leftarrow h_j(\boldsymbol{x}_i) - \theta_j \quad i = 1, \dots, n, \quad j = 1, \dots, J$ // centering
MLR $\leftarrow$ multiple linear regression of Y_i on Z_{ij}
$\hat{\mu}_{\text{reg}} \leftarrow$ estimated intercept from MLR
se $\leftarrow$ intercept standard error from MLR
deliver $\hat{\mu}$, se

This algorithm shows how to use linear regression software to do control variate computation. It is ***essential*** to center the control variates. It may be necessary to drop one or more control variates, if they are linearly dependent in the sample.

To minimize (8.31) with respect to β is a least squares problem and the solution vector β satisfies $\mathrm{Var}(h(\boldsymbol{X}))\beta = \mathrm{Cov}(h(\boldsymbol{X}), f(\boldsymbol{X}))$. If the J by J matrix $\mathrm{Var}(h(\boldsymbol{X}))$ is singular, then one of the h_j is a linear combination of the other $J-1$ control variates. There is no harm in deleting that redundant variate. As a result we can assume that the matrix $\mathrm{Var}(h(\boldsymbol{X}))$ is not singular. Then the optimal value of β is

$$\begin{aligned}\beta_{\text{opt}} &= \mathrm{Var}(h(\boldsymbol{X}))^{-1}\mathrm{Cov}(h(\boldsymbol{X}), f(\boldsymbol{X})) \\ &= \big(\mathbb{E}\big([h(\boldsymbol{X}) - \theta][h(\boldsymbol{X}) - \theta]\big)\big)^{-1}\mathbb{E}\big([h(\boldsymbol{X}) - \theta]f(\boldsymbol{X})\big). \end{aligned} \tag{8.32}$$

In applications we ordinarily do not know β_{opt}. The usual way to estimate it is by replacing expectations by sample averages:

$$\hat{\beta} = \left(\frac{1}{n}\sum_{i=1}^{n}\big(h(\boldsymbol{X}_i) - \bar{H}\big)\big(h(\boldsymbol{X}_i) - \bar{H}\big)^{\mathsf{T}}\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\big(h(\boldsymbol{X}_i) - \bar{H}\big)f(\boldsymbol{X}_i). \tag{8.33}$$

Equation (8.33) is the least squares estimate of β_{opt}.

The usual estimate of μ with control variates is $\hat{\mu}_{\hat{\beta}}$. The estimated variance is

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\hat{\beta}}) = \frac{1}{n^2}\sum_{i=1}^{n}\big(f(\boldsymbol{x}_i) - \hat{\mu}_{\hat{\beta}} - \hat{\beta}^{\mathsf{T}}(h(\boldsymbol{x}_i) - \bar{h})\big)^2.$$

Both the estimate, and its standard error $\sqrt{\widehat{\mathrm{Var}}(\hat{\mu}_{\hat{\beta}})}$, can be computed using standard multiple linear regression software. See Algorithm 8.3. The key insight is to treat μ as the intercept in a multiple linear regression relating $f(\boldsymbol{X})$ to predictors $h_j(\boldsymbol{X}) - \theta_j$. The regression formula is $f(\boldsymbol{X}) \approx \mu + (h(\boldsymbol{X}) - \theta)^{\mathsf{T}}\beta$. It is crucial to subtract θ_j from the control variates in order to make $\mu = \mathbb{E}(f(\boldsymbol{X}))$ match the regression intercept.

The error of the regression estimator using $\beta = \hat{\beta}$ is

$$\begin{aligned}\hat{\mu}_{\hat{\beta}} - \mu &= \hat{\mu}_{\hat{\beta}} - \hat{\mu}_{\beta_{\text{opt}}} + \hat{\mu}_{\beta_{\text{opt}}} - \mu \\ &= (\hat{\mu} - \hat{\beta}^{\mathsf{T}}\bar{H} + \hat{\beta}^{\mathsf{T}}\theta) - (\hat{\mu} - \beta_{\text{opt}}^{\mathsf{T}}\bar{H} + \beta_{\text{opt}}^{\mathsf{T}}\theta) + \hat{\mu}_{\beta_{\text{opt}}} - \mu\end{aligned}$$

$$= (\hat{\beta} - \beta_{\mathrm{opt}})^{\mathsf{T}}(\theta - \bar{H}) + \hat{\mu}_{\beta_{\mathrm{opt}}} - \mu. \tag{8.34}$$

The first term in (8.34) is the product of two components of mean zero, while the second term is the error in the unknown optimal regression estimator. The second term has mean zero, but the first does not in general, because the expected value of a product is not necessarily the same as the product of the expected values. As a result, the control variate estimator is usually biased.

Although estimating β from the sample data brings a bias, that bias is ordinarily negligible. Each of the factors $\hat{\beta} - \beta_{\mathrm{opt}}$ and $\bar{H} - \theta$ is $O_p(n^{-1/2})$ so their product is $O_p(n^{-1})$. The second term $\hat{\mu}_{\beta_{\mathrm{opt}}} - \mu$ in (8.34) is of larger magnitude $O_p(n^{-1/2})$. For large n, the first term is negligible while the second term is unbiased. On closer inspection, the first term in (8.34) is the sum of J contributions, so the bias might be regarded as a J/n term. Ordinarily J is not large enough to cause us to change our mind about whether the sum of J terms of size $O_p(n^{-1})$ is negligible compared to a single $O_p(n^{-1/2})$ term. Thus, for applications with $J \ll \sqrt{n}$, it is common to neglect the bias from using estimated control variate coefficients.

When an unbiased estimator is required, then we can get one by using an estimate of β_{opt} that is independent of the $\boldsymbol{X}_i$ used in $\hat{\mu}_{\hat{\beta}}$. For example $\widetilde{\beta}$ can be computed from (8.33) using only a pilot sample $\widetilde{\boldsymbol{X}}_1, \ldots, \widetilde{\boldsymbol{X}}_m \overset{\text{iid}}{\sim} p$ independent of the $\boldsymbol{X}_i$. Then $\hat{\mu}_{\widetilde{\beta}}$ can be computed by (8.30) using $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ and taking $\beta = \widetilde{\beta}$. Now $\mathbb{E}(\hat{\mu}_{\widetilde{\beta}}) = \mu$ and

$$\mathrm{Var}(\hat{\mu}_{\widetilde{\beta}}) = \mathbb{E}(\mathrm{Var}(\hat{\mu}_{\widetilde{\beta}} \mid \widetilde{\boldsymbol{X}}_1, \ldots, \widetilde{\boldsymbol{X}}_m)) = \frac{1}{n}\mathbb{E}(\sigma^2_{\widetilde{\beta}}).$$

If $f(\boldsymbol{X}_i)$ and $h(\boldsymbol{X}_i)$ have finite fourth moments then $\widetilde{\beta} = \beta_{\mathrm{opt}} + O_p(1/\sqrt{m})$. Since σ^2_β is differentiable with respect to β and takes its minimum at β_{opt} we have $\sigma^2_{\widetilde{\beta}} = \sigma^2_{\beta_{\mathrm{opt}}} + O_p(1/m)$. Exercise 8.20 asks you to allocate computation between the m pilot observations and the n followup observations. See page 35 of the end notes for more sophisticated bias removal.

Example 8.4 (Post-stratification)**.** Suppose that we have strata $\mathcal{D}_1, \ldots, \mathcal{D}_J$ as in §8.4, but instead of a stratified sample, we take $\boldsymbol{X}_i \overset{\text{iid}}{\sim} p$ for $i = 1, \ldots, n$. Let $h_j(\boldsymbol{x}) = \mathbb{1}\{\boldsymbol{x} \in \mathcal{D}_j\}$ for $j = 1, \ldots, J$. The stratum probabilities $\omega_j \equiv \mathbb{P}(\boldsymbol{X} \in \mathcal{D}_j)$ are known. Therefore we can use $h_j(\boldsymbol{x})$ as a control variate with mean $\theta_j = \omega_j$. If we use the stratum indicators as control variates, we get the same estimate $\hat{\mu}_{\mathrm{strat}}$ as in post-stratification. The corresponding variance estimate is slightly different.

Using control variates multiplies the asymptotic variance of $\hat{\mu}$ by a factor $1 - R^2$ where the R^2 is the familiar proportion of variance explained coefficient from linear regression. If $J = 1$ then $R^2 = \rho^2$ where ρ is the correlation of $f(\boldsymbol{X})$ and $h(\boldsymbol{X})$.

If the cost of computing h is high, then the variance reduction from control variates may need to be large in order to make it worthwhile. Let c_f be the cost

of computing $f(\boldsymbol{X})$ including the cost of computing $\boldsymbol{X}$. Let c_h be the additional cost of computing the vector $h(\boldsymbol{X})$ given that we are already committed to computing $\boldsymbol{X}$ and $f(\boldsymbol{X})$. If some parts of the f computation can be saved and reused in computing h, then the related costs should be included in c_f but not in c_h. The cost of computing $\hat{\beta}$ has an $O(J^3)$ term and an $O(nJ^2)$ term. We suppose that the part that grows proportionally to n is included in c_h unless somehow it was needed for computing $f(\boldsymbol{X})$. We also suppose that $J \ll n$, so that the $O(J^3)$ cost may be neglected.

Under the assumptions above, using control variates multiplies the variance by $1 - R^2$ but multiplies the cost per observation by $(c_f + c_h)/c_f$. It improves efficiency if

$$(1 - R^2) \times \frac{c_f + c_h}{c_f} < 1.$$

As a simple special case, suppose that $c_h = Jc_f$. After some rearrangement, we find efficiency is improved if $R^2 > J/(J+1)$.

When J is large it will be very hard to have $R^2 > J/(J+1)$. Multiple control variates may still be worthwhile if they are much less expensive than f. Suitable control variates include low order polynomials in the components of $\boldsymbol{X}$. These are either inexpensive to compute, or nearly free if we already had to compute them in order to compute $f(\boldsymbol{X})$. When the control variates cost on average ϵ times as much as f, then they improve efficiency if $R^2 > J\epsilon/(J\epsilon+1)$.

8.10 Moment matching and reweighting

When we know the value of $\mathbb{E}(\boldsymbol{X}) \equiv \theta$ we can use it to improve our estimate of $\mu = \mathbb{E}(f(\boldsymbol{X}))$ via control variates as described in §8.9. A simple and very direct alternative approach is to adjust the sample values, setting

$$\widetilde{\boldsymbol{X}}_i = \boldsymbol{X}_i + \theta - \bar{\boldsymbol{X}} \tag{8.35}$$

where $\bar{\boldsymbol{X}} = (1/n)\sum_{i=1}^n \boldsymbol{X}_i$, and then estimate μ by the ***moment matching*** estimator

$$\hat{\mu}_{\mathrm{mm}} = \frac{1}{n}\sum_{i=1}^n f(\widetilde{\boldsymbol{X}}_i). \tag{8.36}$$

Moment matching can also be applied to the variance of $\boldsymbol{X}$. Suppose that we know $\mathbb{E}((\boldsymbol{X}-\theta)(\boldsymbol{X}-\theta)^{\mathsf{T}}) \equiv \Sigma$, as we would for a simulation based on $\boldsymbol{X}_i \sim \mathcal{N}(\theta, \Sigma)$. Let $\widehat{\Sigma} = (1/n)\sum_{i=1}^n (\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^{\mathsf{T}}$ be the sample variance matrix, and suppose that $\widehat{\Sigma}$ has full rank, as it will for large enough n, if Σ has full rank. We can then set

$$\widetilde{\boldsymbol{X}}_i = \theta + \Sigma^{1/2}\widehat{\Sigma}^{-1/2}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})$$

and use (8.36).

In financial applications a multiplicative form of moment matching is commonly used replacing geometric Brownian motion sample paths $X_i(t)$ by

$$\widetilde{X}_i(t) = X_i(t) \times \frac{\mathbb{E}(X_i(t))}{\bar{X}(t)}, \quad \text{where} \quad \bar{X}(t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t).$$

An analysis in Boyle et al. (1997) shows that moment matching is asymptotically like using the known moments in control variates but with a non-optimal value for the coefficient β.

It is harder to get confidence intervals for moment matching estimators. The n values $\widetilde{\boldsymbol{X}}_i$ are no longer independent. To get a variance estimate we can repeat the computation K times independently getting $\hat{\mu}_{\mathrm{mm},1}, \ldots, \hat{\mu}_{\mathrm{mm},K}$, and then use

$$\hat{\mu}_{\mathrm{mm}} = \frac{1}{K}\sum_{k=1}^{K} \hat{\mu}_{\mathrm{mm},k}, \quad \text{and}$$

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{mm}}) = \frac{1}{K(K-1)}\sum_{k=1}^{K} (\hat{\mu}_{\mathrm{mm},k} - \hat{\mu}_{\mathrm{mm}})^2.$$

The pooled estimate $\hat{\mu}_{\mathrm{mm}}$ ordinarily has a small bias.

Despite their lesser accuracy and greater complexity, a motivation to use moment matching arises in financial valuation, where the expectations correspond to various prices. There one reasons that the Monte Carlo must reproduce certain known prices, in order to be credible. If one decides to buy (or sell) securities at a price determined by a Monte Carlo model that is higher (respectively lower) than the market price, then an adversarial trader could exploit that difference.

Another way to meet the goal of moment matching is to reweight the sample. We can replace the equal weight estimator by

$$\sum_{i=1}^{n} w_i f(\boldsymbol{X}_i) \tag{8.37}$$

using the same carefully chosen weights w_i for each function f. The weights should satisfy $\sum_{i=1}^{n} w_i X_i = \theta$ in the case of (8.35) above. They should also satisfy $\sum_{i=1}^{n} w_i = 1$.

It turns out that control variate estimates of μ already take the form (8.37). Suppose that the vector h of control variates has $\mathbb{E}(h(\boldsymbol{X})) = \theta \in \mathbb{R}^J$. The case (8.35) simply has $h(\boldsymbol{X}) = \boldsymbol{X}$. Then the estimator (8.33) of β takes the form

$$\hat{\beta} = \sum_{i=1}^{n} S_{HH}^{-1}(h(\boldsymbol{X}_i) - \bar{H}) f(\boldsymbol{X}_i)$$

for $S_{HH}^{-1} = \sum_{i=1}^{n} (h(\boldsymbol{X}_i) - \bar{H})(h(\boldsymbol{X}_i) - \bar{H})^{\mathsf{T}}$. As a result the control variate estimator is

$$\hat{\mu}_{\hat{\beta}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i) - \hat{\beta}^{\mathsf{T}}(\bar{H} - \theta) = \sum_{i=1}^{n} w_i f(\boldsymbol{X}_i), \quad \text{for}$$

$$w_i = \frac{1}{n} - (h(\boldsymbol{X}_i) - \bar{H})^{\mathsf{T}} S_{HH}^{-1}(\bar{H} - \theta).$$

One slim advantage of moment matching over control variates is that it will automatically obey some natural constraints. For example, if $f(\boldsymbol{x}) = \exp(\boldsymbol{x})$ then we know that $\mathbb{E}(f(\boldsymbol{X}))$ cannot be negative. It is possible for control variates to supply a negative estimate for such a quantity that must be positive. By contrast, we can be sure that $\hat{\mu}_{\mathrm{mm}}$ is not negative when $f(\boldsymbol{x}) \geqslant 0$ always holds. Some methods to find non-negative weights with $\sum_i w_i h(\boldsymbol{X}_i) = \theta$ and $\sum_i w_i = 1$ (when they exist) are describe on page 38 of the end notes.

Moment matching and related methods allow one to bake in certain desirable properties of the sample points $\widetilde{\boldsymbol{X}}_i$. Their main attraction arises when those properties are important enough to give up on some estimation accuracy and simplicity of forming confidence intervals.

Chapter end notes

There is a large literature on variance reduction methods. For surveys, see Wilson (1984) and L'Ecuyer (1994).

Antithetic sampling was introduced by Hammersley and Morton (1956). Some generalizations of antithetic sampling are considered in Chapter 10.

Stratification is a classic survey sampling method. See Cochran (1977), for issues of variance estimation and also for design of strata. It is not just stratification. Antithetics, control variates and importance sampling (Chapter 9) have direct antecedents in the survey sampling literature.

The difference estimator is also commonly used in classical quadrature methods. Suppose that both $h(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are unbounded, but $f(\boldsymbol{x}) - h(\boldsymbol{x})$ is bounded, and $\int h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is known. Then it often pays to use numerical quadrature on $f - h$ and add in the known integral of h. For Monte Carlo sampling it will ordinarily be better to use regression estimator. However for quasi-Monte Carlo and randomized quasi-Monte Carlo (Chapters 15 through 17) we may prefer the difference estimator if $f - h$ is then of bounded variation.

The ratio and product estimators are not available when $\theta = \mathbb{E}(h(\boldsymbol{X})) = 0$. Their typical applications are in problems where $h(\boldsymbol{x}) > 0$. The reason that complicated nonlinear control variates are seldom used is that, in large samples, they are almost equivalent to the regression estimator, which is simple to use. See Glynn and Whitt (1989).

The regression estimator for control variates has a mildly annoying bias. Avramidis and Wilson (1993) describe a way to get rid of it. They split the sample into $m \geqslant 2$ subsets of equal size and arrange that each coefficient estimate $\hat{\beta}$ is always applied to points independent of it. The result is an unbiased estimate of μ using control variates. When $m \geqslant 3$ they are also able to get an unbiased estimate of $\mathrm{Var}(\hat{\mu})$.

Kahn and Marshall (1953) make an early mention of the method of common random numbers, referring to it as correlation of samples. They liken it to

pairing and blocking which had long been an important part of the design of physical experiments.

Lunney and Anderson (2009) use Monte Carlo methods to measure the power of the content uniformity test under some alternatives with non-normally distributed data.

Asmussen and Glynn (2007, Chapter VII) cover Monte Carlo estimation of derivatives. They include many algorithms of varying complexity for the case where $\boldsymbol{X}$ is a process and θ is a parameter of that process. Burgos and Giles (2012) look at multilevel Monte Carlo for estimation of derivatives.

Hesterberg and Nelson (1998) explore the use of control variates for quantile estimation. For random pairs (X_i, Y_i) one or more known quantiles of the X distribution can be used as control variates for α quantile of the Y distribution. The most direct approach is to estimate $\mathbb{E}(\mathbb{1}_{Y\leqslant y})$ using $\mathbb{1}_{X\leqslant x_1}, \dots, \mathbb{1}_{X\leqslant x_s}$ as control variates, and estimate the α quantile of Y to be the value y for which $\widehat{\mathbb{E}}(\mathbb{1}_{Y\leqslant y}) = \alpha$. They consider using a small number of values x_j at or near the α quantile of X. Extreme variance reductions are hard to come by because it is hard to find regression variables that are extremely predictive of a binary value like $\mathbb{1}_{Y\leqslant y}$.

Barraquand (1995) and Duan and Simonato (1998) use some moment matching methods on sample paths of geometric Brownian motion. Cheng (1985) gives an algorithm to generate n random vectors from the distribution $\mathcal{N}(0, I_p)$ conditionally on their sample mean being μ and sample covariance being Σ. In that approach the constraints are built in to the sample generation rather than imposed by transformation afterwards. Pullin (1979) had earlier done this for samples from $\mathcal{N}(0, 1)$.

Common random numbers with randomness in h

Here we allow the function $h(\boldsymbol{X}, \theta)$ in common random numbers to generate further random numbers. We assume that the number of further random numbers $h(\boldsymbol{X}, \theta)$ uses depends on both $\boldsymbol{X}$ and θ. If instead $h()$ always takes the same number of uniform random numbers we can include them in $\boldsymbol{X}$ and proceed as if h does not generate random variables.

We begin with Algorithm 8.1 and we assume that all the dependence we wanted to incorporate comes through the shared $\boldsymbol{X}_i$ and so $h(\boldsymbol{X}_i, \theta_j)$ for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$ are conditionally independent given $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$.

In Algorithm 8.1, an h that consumes random numbers would advance the random number stream by some number of positions and thereby change $\boldsymbol{X}_2, \dots, \boldsymbol{X}_n$. The differences $\hat{\mu}_j - \hat{\mu}_k$ would still be unbiased estimates of $\mu_j - \mu_k$. The additional randomness in h would increase the variance of $\hat{\mu}_j - \hat{\mu}_k$, reducing the gain from common random numbers. Because the n sample differences $h(\boldsymbol{X}_i, \theta_j) - h(\boldsymbol{X}_i, \theta_k)$ going into that estimate are still statistically independent, our confidence intervals remain reliable.

The challenge with Algorithm 8.1 starts when we consider changing our parameter list $\theta_1, \dots, \theta_m$, perhaps by adding $\theta_{m+1}, \dots, \theta_{m+k}$. To account for changing parameters it is less ambiguous to write $\hat{\mu}(\theta_j)$ instead of $\hat{\mu}_j$. When

h consumes random numbers, then changing the parameter list $\theta_1, \ldots, \theta_m$, can change $\boldsymbol{X}_2$ and all subsequent $\boldsymbol{X}_i$ that Algorithm 8.1 uses.

If we add new parameters $\theta_{m+1}, \ldots, \theta_{m+k}$ to our list and rerun Algorithm 8.1 for all $m+k$ parameter values, then it is likely that all of our old estimates $\hat{\mu}(\theta_j)$ for $j \leqslant m$ will have changed. The estimates still reflect common random numbers. But we might have preferred those old values to remain fixed.

A very serious problem (i.e., an error) arises when we store the values $\hat{\mu}(\theta_j)$ for $j = 1, \ldots, m$, and then instead of re-running Algorithm 8.1 on the whole list, we just run it on the list of k new parameter values. Then the new estimates $\hat{\mu}(\theta_{m+1}), \ldots, \hat{\mu}(\theta_{m+k})$ will not have been computed with the same $\boldsymbol{X}_i$ that the old ones used. Even though that algorithm starts by setting the seed, synchronization will already be lost for $\boldsymbol{X}_2$ because h generated random numbers. We would have lost the accuracy advantage of common random numbers for comparisons involving one of the first m parameters and one of the last k. Also, some of the random numbers used to generate $\boldsymbol{X}_i$ for the first set of parameters may end up incorporated into both $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$ (or some other set of variables) for the second set. The differences $h(\boldsymbol{X}_i, \theta_r) - h(\boldsymbol{X}_i, \theta_s)$, $i = 1, \ldots, n$ would not be independent if $r \leqslant m < s$. So we would get unreliable standard deviations for those comparisons.

To be sure that $\boldsymbol{X}_i$ is the same for all sets Θ, we should not let h use the same stream of random numbers that $\boldsymbol{X}_i$ are generated from. Even giving h its own stream of random numbers leaves us with synchronization problems. Computing $h(\boldsymbol{X}_1, \theta_{m+1})$ would affect the random numbers that $h(\boldsymbol{X}_2, \theta_1)$ sees.

If we want $\hat{\mu}(\theta_j)$ to be unaffected by the other $\theta_k \in \Theta$, then the solution is to give h a different random number stream for each value of θ that we use. One approach is to maintain a lookup table of θ's and their corresponding seeds. Another is to hash the value of θ_j into a seed (or a stream identifier) for h to use. If each θ_j gets its own stream, as in L'Ecuyer et al. (2002) then the common seed for all of those streams gets set at the beginning of the algorithm. If each θ_j is hashed into its own seed for a random number generator like the Mersenne Twister (Matsumoto and Nishimura, 1998), then seeded copies of that generator should be created at the beginning of the algorithm. Now each $\hat{\mu}(\theta)$ is a reproducible function of θ and n and the seeds used.

Now consider Algorithm 8.2 where h consumes random numbers. For each θ_j it sets the seed then does a Monte Carlo sample. It is more fragile than Algorithm 8.1. That algorithm still works if we run all θ_j at once and do not mind having $\hat{\mu}(\theta_j)$ depend on the set of other θ values in Θ. For Algorithm 8.2, if h generates random numbers then $\boldsymbol{X}_i$ for $i \geqslant 2$ will vary with θ_j and we lose synchronization. To ensure that $\boldsymbol{X}_i$ are really common we should not let h use the stream that we use to generate $\boldsymbol{X}_i$. To keep each $\hat{\mu}(\theta)$ unaffected by changes to the set Θ, we should once again give every value of θ its own stream, and set the seed for that stream at the same time the $\boldsymbol{X}$ stream's seed is set.

Alternative reweightings

As described in §8.10, control variates reweight the sample values but might include some negative weights. We would prefer to have weights w_i that satisfy

$$w_i \geqslant 0, \quad \sum_{i=1}^n w_i = 1, \quad \text{and} \quad \sum_{i=1}^n w_i h(\boldsymbol{X}_i) = \theta. \tag{8.38}$$

Ideally, the weights w_i should be as close to $1/n$ as possible, subject to the constraints in (8.38). Then we may estimate μ by

$$\hat{\mu}_w = \sum_{i=1}^n w_i f(\boldsymbol{X}_i).$$

Constraints (8.38) cannot always be satisfied. If $\min_{1\leqslant i\leqslant n} h_j(\boldsymbol{X}_i) > \theta_j$ then there is no way to satisfy (8.38). More generally, if θ is outside the convex hull of $\{h(\boldsymbol{X}_1), \dots, h(\boldsymbol{X}_n)\}$, so that there exists a hyperplane with $\theta \in \mathbb{R}^J$ on one side and all of $h(\boldsymbol{X}_i)$ on the other, then (8.38) cannot be satisfied. If θ is outside the convex hull of $h(\boldsymbol{X}_i)$ then maybe n is too small, or J is too large, or the functions h_j are poorly chosen.

If a solution to (8.38) exists then there is an $n-J-1$ dimensional family of solutions. To choose weights in this family we need to choose a measure of their distance from $(1/n, \dots, 1/n)$. One such way is to maximize the log empirical likelihood $-\sum_{i=1}^n \log(nw_i)$ subject to (8.38). A second way is to maximize the entropy $-\sum_{i=1}^n w_i \log(w_i)$ subject to (8.38). Both of these criteria favor w_i that are nearly equal. Each of them leads to weighted Monte Carlo estimates with the same asymptotic variance that $\hat{\mu}_{\hat{\beta}}$ has.

If we maximize the empirical likelihood, then a Lagrange multipliers argument yields

$$w_i^{\mathrm{EL}} = \frac{1}{n} \frac{1}{1 + \lambda^{\mathsf{T}}(h(\boldsymbol{X}_i) - \theta)}$$

where the Lagrange multiplier $\lambda \in \mathbb{R}^J$ satisfies

$$\sum_{i=1}^n \frac{h(\boldsymbol{X}_i) - \theta}{1 + \lambda^{\mathsf{T}}(h(\boldsymbol{X}_i) - \theta)} = 0.$$

(Owen, 2001, Chapter 3) gives details including computation of λ. Empirical likelihood and entropy are two members in a family of non-negative weighting methods. For Monte Carlo applications where non-negativity is not needed, regression based control variates are simpler to use.

Exercises

Antithetics

8.1. Given $\epsilon > 0$, construct an increasing function $f(x)$ on $0 \leqslant x \leqslant 1$ such that

$$0 \geqslant \mathrm{Corr}(f(X), f(1-X)) > -\epsilon$$

for $X \sim \mathbf{U}(0,1)$.

8.2. Find an example for the following set of conditions, or prove that it is impossible to do so: $0 < \mathrm{Var}(\hat{\mu}_{\mathrm{anti}}) < \mathrm{Var}(\hat{\mu}) = \infty$. Here $\hat{\mu}$ is ordinary Monte Carlo sampling with a finite even number $n \geqslant 2$ of function values and $\hat{\mu}_{\mathrm{anti}}$ is antithetic sampling with $n/2$ pairs. If this is possible, then $\hat{\mu}_{\mathrm{anti}}$ has an infinite efficiency relative to ordinary Monte Carlo without having 0 variance.

8.3. Show that the correlation in antithetic sampling is

$$\rho = \frac{\sigma_{\mathrm{E}}^2 - \sigma_{\mathrm{O}}^2}{\sigma_{\mathrm{E}}^2 + \sigma_{\mathrm{O}}^2},$$

in the notation of §10.2.

8.4 (Antithetic sampling and spiky integrands)**.** Here we investigate what happens with antithetic sampling and a spiky function. We will use

$$f(x) = \begin{cases} 0, & 0 < x \leqslant 0.9 \\ 100, & 0.9 < x \leqslant 0.91 \\ 0, & 0.91 < x < 1 \end{cases}$$

for $X \sim \mathbf{U}(0,1)$ as a prototypical spiky function.

- **a)** Determine whether antithetic sampling is helpful, harmful, or neutral for the example f. You may do this by finding the variance of $\hat{\mu}$ under IID and under antithetic sampling using the same sample size. You may find the variances either theoretically or from a large enough simulation.
- **b)** Explain your findings from the part above, in terms of the even and odd parts of f.
- **c)** Construct a spiky function for which you would have reached a very different conclusion about the effectiveness of antithetic sampling.

Stratification

8.5. Prove equation (8.15), which represents the variance reduction from proportional allocation in terms of a correlation between f and the within stratum mean of f.

8.6. Equation (8.14) expresses the sampling variance of the stratified estimator and the ordinary MC estimator in terms of between and within variances σ_{B}^2 and σ_{W}^2. Given f with $\int f(\boldsymbol{x})^2 \,\mathrm{d}\boldsymbol{x} < \infty$ show how to construct functions $f_{\mathrm{B}}(\boldsymbol{x})$ and $f_{\mathrm{W}}(\boldsymbol{x})$ such that $f(\boldsymbol{x}) = f_{\mathrm{B}}(\boldsymbol{x}) + f_{\mathrm{W}}(\boldsymbol{x})$ with $\int f_{\mathrm{W}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int f_{\mathrm{B}}(\boldsymbol{x}) f_{\mathrm{W}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = 0$ and $\int f_{\mathrm{B}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \mu$, $\int f_{\mathrm{W}}(\boldsymbol{x})^2 \,\mathrm{d}\boldsymbol{x} = \sigma_{\mathrm{W}}^2$, $\int (f_{\mathrm{B}}(\boldsymbol{x}) - \mu)^2 \,\mathrm{d}\boldsymbol{x} = \sigma_{\mathrm{B}}^2$, and for which the stratified sampling estimate of the mean of f_{B} has variance zero.

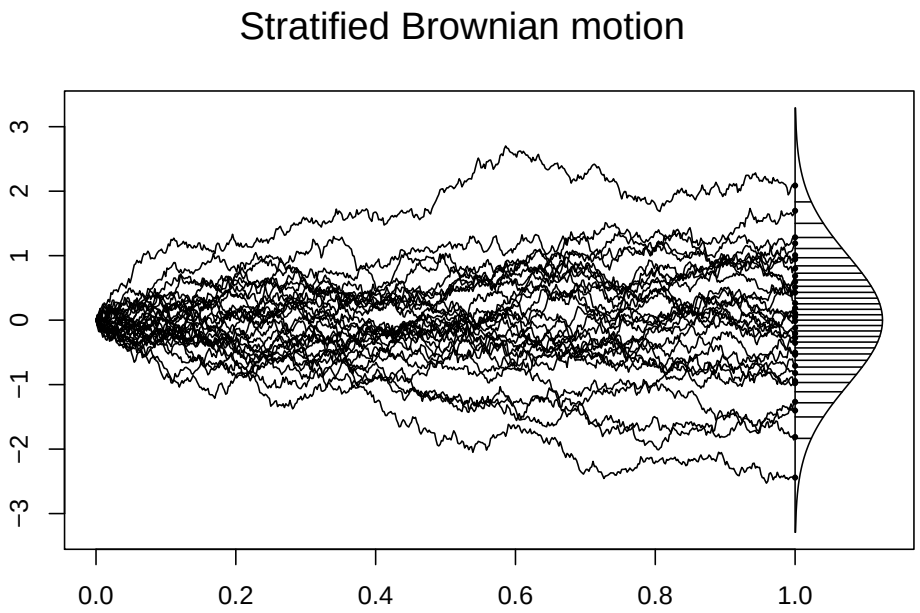

Figure 8.6: This figure shows 30 sample paths of standard Brownian motion $B(\cdot) \sim \mathsf{BM}(0,1)$ on $\mathcal{T} = [0,1]$. They are stratified on $B(1) \sim \mathcal{N}(0,1)$. See Exercise 8.7. Also shown is the $\mathcal{N}(0,1)$ density function partitioned into 30 equi-probable intervals.

8.7 (Stratified Brownian motion)**.** Here we investigate stratified Brownian motion, as shown in Figure 8.6. Let path i at time t take the value $B_i(t)$ for $i = 1, \ldots, N$ and $t \in \{1/M, 2/M, \ldots, 1\}$. To stratify standard Brownian motion on its endpoint, we take $B_i(1) = \Phi^{-1}((i-U_i)/N)$ for independent $U_1, \ldots, U_N \sim \mathbf{U}(0,1)$. Points $B_i(j/M)$, for $j = 1, \ldots, M-1$ are then sampled conditionally on $B_i(1)$. See §xxx.

a) Write a function to generate stratified standard Brownian motion. It should take arguments $M, N \in \mathbb{N}$, and $i \in \{1, \ldots, N\}$. It should produce the sample path of stratified $B_i(t)$ at $t = j/M$ for $j = 1, \ldots, M$. Describe how you sampled the path $B_i(\cdot)$, conditionally on $B_i(1)$, with enough detail to make it clear that your method is correct. Turn in your code with comments. [Note: if you prefer, you may instead write the function to generate and return all N paths $i = 1, \ldots, N$ at once.]

b) Generalize your function to generate stratified Brownian motion with drift $\delta \in \mathbb{R}$ and volatility $\sigma \geqslant 0$ on the interval $\mathcal{T} = [0,T]$ for $T > 0$. As before the value of $B(T)$ is stratified. Explain how your generalization works, and turn in your code. You may either pass the new arguments δ, σ, and T into a generalized version of your previous function, or you may write a wrapper function that calls your previous function and modifies its output

to take account of δ, σ and T.

c) Let $S(\cdot) \sim \mathsf{GBM}(S_0, \delta, \sigma)$ be geometric Brownian motion (§6.4). For $M = 100$, let

$$f(S(\cdot)) = \max_{0 \leqslant j \leqslant M} S(j/M) - \min_{0 \leqslant j \leqslant M} S(j/M).$$

We want $\mu = \mathbb{E}(f(S(\cdot)))$ for $\delta = 0.05$, $\sigma = 0.3$, and $T = 1$. For $N = 1000$ and $M = 100$ generate two independent stratified Geometric Brownian motions with these parameters. Estimate μ and give a 99% confidence interval. [Hint: the two independent stratified samples can be pooled into one stratified sample of $n = 2N$ paths, with $J = N$ strata having $n_j = 2$ for $j = 1, \ldots, N$.]

The function f is related to the value of a lookback option whose payoff is equivalent to buying at the minimum and selling at the maximum price in the time interval $[0, T]$. As given, f omits the discount factor $e^{-\delta T}$ that compensates for waiting until time T to collect the payoff.

d) Estimate the variance reduction obtained from stratification. Use R independent replications of the stratified sampling method on $n = 2N$ paths, where $R \geqslant 300$. The variance should be compared to that obtained by plain Monte Carlo with $2N$ paths.

e) Compare the time required to compute $2N = 2000$ sample paths of length $M = 100$ by stratification to that required to compute $2N$ sample paths of length M without stratification. Report the details of the hardware, operating system, and the software in which you made the comparison.

8.8 (Stratification with $n_j = 1$)**.** Consider proportional allocation (see §8.4) in the special case where all the strata have equal probability. Then $\omega_j = 1/J$ and $n_j = m$ for $j = 1, \ldots, J$ where the sample size is $n = mJ$.

a) Suppose first that $m \geqslant 2$ and let s_j^2 be as given in (8.10). Define $\bar{s^2} = (1/J)\sum_{j=1}^{J} s_j^2$. Show that the formula for $\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat}})$ in (8.10) reduces to $\bar{s^2}/n$.

b) Now suppose that $m = 1$ and that $n = J$ is an even number. We saw in §xxx that the stratified sampling estimate $\hat{\mu}_{\mathrm{strat}}$ is $\bar{Y} = (1/n)\sum_{i=1}^{n} Y_i$, in this setting where $Y_i = f(X_i)$. For $m = 1$ we cannot use equation (8.10) for $\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat}})$. For $j = 1, \ldots, n/2$ let $\widetilde{s}_j^2 = (f(X_{2j-1}) - f(X_{2j}))^2$, put $\overline{\widetilde{s^2}} = (2/n)\sum_{j=1}^{n/2} \widetilde{s}_j^2$ and let $\widetilde{V} = \overline{\widetilde{s^2}}/n$. Prove that $\mathbb{E}(\widetilde{V}) \geqslant \mathrm{Var}(\bar{Y})$.

c) Suppose now that stratum i is $[(i-1)/n, i/n)$, that n is very large, and f has two derivatives on $[0, 1]$. Roughly how large will $\mathbb{E}(\widetilde{V})/\mathrm{Var}(\bar{Y})$ be?

Conditioning

8.9. Let $(X, Y) \sim \mathbf{U}(0, 1)^2$ and put $f(x, y) = e^{g(x)y}$ for $g(x) = \sqrt{5/4 + \cos(2\pi x)}$. Let $h(x) = (e^{g(x)} - 1)/g(x)$.

a) Using $n = 10^6$ samples estimate the variance of $f(x, y)$. Similarly, estimate the variance of $h(x)$.

b) Report the efficiency gain from conditioning assuming that f and h cost the same amount of computer time. Then report the efficiency gain taking account of the time it takes to compute both f and h. In this case give details of the computing environment that you obtained the results for. Also hand in your source code.

c) Repeat the two steps above for $g(x) = \sqrt{1 + \cos(2\pi x)}$ taking special care near $x = 1/2$. (Hint: you may need a Taylor expansion.)

Exercises 8.10 through 8.13 require a function that computes the CDF of the Gamma distribution.

8.10. Here we find the answer to the roulette problem of §8.8, using conditional Monte Carlo, but no other variance reductions.

a) What is the numerical value of α_{19} for wheel 1?

b) Use conditional Monte Carlo to find the probability that number 19 has the highest probability of coming up on wheel 1 of §8.8. Give a 99% confidence interval.

c) Estimate the probability that 3 is the highest probability number for wheel 2 of Table 8.3 and give a 99% confidence interval.

d) Give a 99% confidence interval for p_{19} of wheel 1 and p_3 of wheel 2. A gambler will make money in the long run by betting on a wheel with $p > 1/36$, and lose if $p < 1/36$, while the game is fair if $p = 1/36$. Do these confidence intervals include $1/36$? You don't need to do a Monte Carlo for this part, the Monte Carlo you need is reported in Table 8.3.

e) On wheel 1, the second most common number was 36. Estimate the probability that number 36 is the most probable, and give a 99% confidence interval.

8.11. Devise a strategy to find the probability that number 19 is the ***second best*** number for wheel 1 based on the data in Table 8.3. Give a formula for your method, and implement it, reporting the answer and a 99% confidence interval.

8.12. For the simulation in Exercise 8.10b estimate how much the variance was reduced by conditioning.

8.13. For the simulation in Exercise 8.10b sample p_{19} by stratified sampling, with 2 observations per stratum and the same sample size you used there (plus one if your sample size was odd). Report the ratio of the estimated variance of $\hat{p}_{19}$ using ordinary IID sampling to that using stratified sampling. Both Monte Carlos in the ratio should employ conditioning.

8.14. In introductory probability exercises we might imagine a perfect roulette wheel with $p_j = 1/38$ exactly. In Exercise 8.10 we considered $\boldsymbol{p}$ uniformly distributed over all possible probability vectors. Neither of these models is reasonable. A more plausible model is that $\boldsymbol{p} \sim \mathrm{Dir}(A, A, \ldots, A)$ for some value of A with $1 < A < \infty$. Then $\boldsymbol{p} \mid \boldsymbol{X} \sim \mathrm{Dir}(A + C_1, \ldots, A + C_{38})$.

a) For what value of A does

$$\mathbb{E}\Big(\sum_{j=1}^{38}(p_j - 1/38)^2\Big) = \sum_{j=1}^{38}(C_j/N - 1/38)^2$$

hold, where the counts C_j come from wheel j, and $N = \sum_{j=1}^{38} C_j$?

b) Consider the following empirical Bayes analysis. Taking the number A obtained from part **a** replace the prior $\mathrm{Dir}(1,\dots,1)$ by $\mathrm{Dir}(A,\dots,A)$. This empirical Bayes analysis will change the estimated probability that 19 is really the best hole for wheel 1. Assuming that $A > 1$, the prior for $\boldsymbol{p}$ will concentrate closer to the center of the simplex, and we anticipate a lower probability that wheel 19 is best.

How much does $\mathbb{P}(p_{19} \geqslant \max_{1\leqslant j\leqslant 38} p_j)$ change when we replace $\alpha_j = 1$ by $\alpha_j = A$ in the prior distribution? Use conditional Monte Carlo and common random numbers with $n = 10{,}000$ sample points to estimate the difference in these probabilities.

Control variates

8.15. Let f and h be two functions of the random variable $\boldsymbol{X} \sim p$. Define $\mu = \mathbb{E}(f(\boldsymbol{X}))$, $\theta = \mathbb{E}(h(\boldsymbol{X}))$, and $\Delta = \mu - \theta$. Assume that we know θ and that our goal is to estimate Δ. Two estimators come to mind. The first estimator is $\hat{\Delta}_1 = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - \theta)$. The second estimator, $\hat{\Delta}_2$ is obtained by estimating the mean of $f(\boldsymbol{X}) - h(\boldsymbol{X})$, using $h(\boldsymbol{X})$ as a control variate.

For which values of $\rho = \mathrm{Corr}(f(\boldsymbol{X}), h(\boldsymbol{X}))$ is $\hat{\Delta}_2$ more efficient than $\hat{\Delta}_1$? You may use the following simplifying assumptions:

i) $\mathrm{Var}(f(\boldsymbol{X})) = \mathrm{Var}(h(\boldsymbol{X})) = \sigma^2 \in (0,\infty)$.
ii) The cost to evaluate h is the same as that for f.
iii) The cost to sample $\boldsymbol{X}$ is negligible.
iv) n is large enough that the delta method approximation to the variance of the regression estimator is accurate enough.

8.16. In quadrature problems it is common to subtract a singularity that we can handle analytically. Here we look at what might happen if we used control variates instead.

Let $f(x) = x^{-1/2} + x$ for $x \in (0,1)$. Let $h(x) = x^{-1/2}$. We know that $\theta \equiv \int_0^1 h(x)\,\mathrm{d}x = 2$, and of course $\mu \equiv \int_0^1 f(x)\,\mathrm{d}x = 5/2$. Suppose that $X \sim \mathbf{U}(0,1)$. Here we estimate $\mathbb{E}(f(X))$ by Monte Carlo using h as a control variate, and forgetting for the moment that we know μ. That is, we use $\hat{\mu}_{\hat{\beta}}$ instead of $\hat{\mu}_1 = (1/n)\sum_{i=1}^n (f(x_i) - 1(h(x_i) - 2))$.

a) Show that $\mathrm{Var}(f(X) - \beta h(X)) < \infty$ if and only if $\beta = 1$. State the variance of $f(X) - h(X)$.

b) Let $\hat{\mu}_{\hat{\beta}}$ be the usual control variate estimate of μ. Suppose that $n = 1000$. Do a nested Monte Carlo analysis that repeats the size n simulation $R =$

10,000 times. Report the sample mean, sample variance and histogram of $\hat{\beta}$ over the R replicates. Does $\hat{\beta}$ look like it is roughly normally distributed around the true value $\beta = 1$?

c) Show the sample mean, sample variance and histogram of $\hat{\mu}_{\hat{\beta}}$ over the R estimates. Compare $\hat{\mu}_{\hat{\beta}}$ to $\hat{\mu}_2$, by judging their sample squared errors. For practical purposes, do they appear to have very similar or sharply different accuracy? Either way, which one came out better than the other, in your simulations?

d) Repeat the previous two parts with $R = 10{,}000$ and $n = 50$.

e) Inspect the histogram of $\hat{\beta}$ values from part b. Find an apparent upper bound $\hat{\beta} \leqslant A$ and then prove it holds. [Hint: Chebyshev's sum inequalities may be useful. If $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n$ and $b_1 \geqslant b_2 \geqslant \cdots \geqslant b_n$ and $c_1 \leqslant c_2 \leqslant \cdots \leqslant c_n$ then $n\sum_i a_i b_i \geqslant \sum_i a_i \sum_i b_i$ and $n\sum_i a_i c_i \leqslant \sum_i a_i \sum_i c_i$.]

8.17. Suppose that $\mathbb{E}(f(\boldsymbol{X})^2) < \infty$ and $\mathbb{E}(h(\boldsymbol{X})^2) < \infty$ and $\theta = \mathbb{E}(h(\boldsymbol{X})) \neq 0$. Consider the ratio estimator $\hat{\mu}_R = \theta \sum_{i=1}^n f(\boldsymbol{X}_i) / \sum_{i=1}^n h(\boldsymbol{X}_i)$. Show that $\mathbb{P}(|\hat{\mu}_R - \mu| > \epsilon) \to 0$ holds for any $\epsilon > 0$, and $\mu = \mathbb{E}(f(\boldsymbol{X}))$.

8.18. Under the conditions of Exercise 8.17, show that $\mathbb{P}(|\hat{\mu}_P - \mu| > \epsilon) \to 0$, where $\hat{\mu}_P = \Big(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{X}_i)\Big)\Big(\frac{1}{n}\sum_{i=1}^n h(\boldsymbol{X}_i)\Big)/\theta$.

8.19. Suppose that a control variate $g(\boldsymbol{X})$ has a correlation of 0.1 with the variable $f(\boldsymbol{X})$ of interest. By how much does its use reduce the variance of $\mathbb{E}(f(\boldsymbol{X}))$? How much faster than f does the control variate function have to be for its use to improve the efficiency measure (8.1)?

8.20. For the unbiased control variate problem suppose that we will take $N = n + m$ observations. The fraction of the sample allocated to finding the pilot estimate $\widetilde{\beta}$ is $f = m/N$. Then a fraction $1-f$ is used for the final estimate. Suppose that the mean squared error takes the form $(1/n)(A+\sigma_0^2/m)$ for constants $A > 0$ and $\sigma_0^2 > 0$.

a) Find the value of f that minimizes the mean squared error for fixed $N > 0$ over the interval $0 \leqslant f \leqslant 1$. Let f vary continuously, even though fN must really be an integer.

b) Let $m(N)$ be the optimal solution from part **a**. For what r, if any, does $m(N)/N^r$ approach a limit as $N \to \infty$?

If the answer in part **b** is $r = 1$ then the pilot sample should be a fixed fraction of the total data set. For $r = 0$ we get a fixed number of pilot samples.

8.21. If $\hat{\mu}$ and $\hat{\theta}$ are positively correlated then $\hat{\mu}/\hat{\theta}$ should be more stable because fluctuations in the numerator and denominator will offset each other. If they are negatively correlated we would expect $\hat{\mu}\hat{\theta}$ to be more stable. Investigate this intuition by finding the delta method approximation to the variance of $\hat{\mu}_R$ and $\hat{\mu}_P$. Assume that $0 < \mathrm{Var}(f(\boldsymbol{X})) = \sigma^2 < \infty$, $0 < \mathrm{Var}(\hat{\theta}) = \tau^2 < \infty$, $\mathrm{cor}(f(\boldsymbol{X}), h(\boldsymbol{X})) = \rho \in (-1, 1)$, and that $\theta \neq 0$. By comparing the variances, decide whether $\rho > 0$ favors the product estimator, or the ratio estimator, or neither as $n \to \infty$.

Common random numbers

8.22. The content uniformity test on page 18 involved a small shift of the target value from 100 towards $\bar{x}$, but not going more than a distance of 1.5 units. This was implemented by the target shifting function $M(x)$ in equation (8.22). It is natural to wonder whether target shifting makes much difference to the acceptance probability. We can turn off that feature by replacing $M(x)$ with $\widetilde{M}(x) = 100$ for all x. Assume throughout that $X_j \sim \mathcal{N}(\mu, \sigma^2)$ for $j = 1, \ldots, 30$ are independent.

a) Suppose that $\mu = 102$ and $\sigma = 3$. Estimate the amount (and direction) of the change in acceptance probability that arises from the use of target shifting.

b) Now suppose that $\mu = 100$. Is there any σ for which target shifting changes the acceptance probability by more than 5%?

c) Are there any (μ, σ) pairs for which the acceptance probability changes by more than 50% due to target shifting? If so, describe the region where this happens. If not, what is the greatest change one can find? In either case, indicate which (μ, σ) pairs result in the greatest change in acceptance probability.

Make a reasonable choice of Monte Carlo method for this problem, explaining the reasons for your choice. State the sample size you used. There will necessarily be numerical uncertainty because you cannot sample all configurations and n must be bounded.

8.23. In the content uniformity test, a really good product will pass at the first level, while a very bad one will not pass at all. Which combinations of μ and σ lead to the greatest probability that the test will have to carry on to the second level, but will then pass?

8.24. Figure 8.5 was made with $n = 100{,}000$ simulated cases, which may have been more than necessary. How could one determine whether a given sample size n is large enough for such a contour plot?

8.25. In financial applications one often needs the partial derivatives of an option value with respect to parameters like δ and σ. These derivatives, termed 'Greeks' are needed for hedging. For the lookback option function f of Exercise 8.7c define $g(\delta, \sigma, T) = \mathbb{E}(f(S(\cdot)))$ for the given values of δ, σ, and T. Using plain Monte Carlo, without stratification, estimate the following:

a) $g(0.051, 0.3, 1) - g(0.05, 0.3, 1)$,

b) $g(0.05, 0.31, 1) - g(0.05, 0.3, 1)$, and

c) $g(0.05, 0.3, 1.01) - g(0.05, 0.3, 1)$.

Give a confidence interval in each case. Make a reasonable choice for n.

8.26. Give an example where common random numbers increases variance. That is, find a distribution p and functions f and g and prove that $\mathrm{Var}(\widehat{D}_{\mathrm{com}}) > \mathrm{Var}(\widehat{D}_{\mathrm{ind}})$ holds with your p, f and g.

8.27. Exercise 5.13 is about sampling a bivariate distribution with Gaussian margins and the same copula that the Marshall-Olkin bivariate exponential distribution has.

In the notation of that exercise, suppose that $\lambda_1 = \lambda_2 = 1$ and that we want $\mathrm{Corr}(Y_1, Y_2) = 0.7$.

a) What value of λ_3 should we use? Devise a way to solve this problem using common random numbers and a fixed $n \times 3$ matrix with independent components that were sampled from the $\mathbf{U}(0,1)$ distribution. Report the value of λ_3 that you get.

b) Repeat the previous part 10 times independently and report the 10 values you get.

Bibliography

Anderson, D. F. and Higham, D. J. (2012). Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics. *Multiscale Modeling & Simulation*, 10(1):146–179.

Asmussen, S. and Glynn, P. W. (2007). *Stochastic simulation*. Springer, New York.

Avramidis, A. N. and Wilson, J. R. (1993). A splitting scheme for control variates. *Operations Research Letters*, 14:187–198.

Barraquand, J. (1995). Numerical valuation of high dimensional multivariate European securities. *Management Science*, 41(12):1882–1891.

Boyle, P. P., Broadie, M., and Glasserman, P. (1997). Monte Carlo methods for security pricing. *Journal of economic dynamics and control*, 21(8):1267–1321.

Burgos, S. and Giles, M. B. (2012). Computing Greeks using multilevel path simulation. In *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 281–296. Springer.

Cheng, R. C. H. (1985). Generation of multivariate normal samples with given sample mean and covariance matrix. *Journal of Statistical Computation and Simulation*, 21(1):39–49.

Cochran, W. G. (1977). *Sampling Techniques (3rd Ed)*. John Wiley & Sons, New York.

Duan, J.-C. and Simonato, J. (1998). Empirical martingale simulation for asset prices. *Management Science*, 44(9):1218–1233.

Glynn, P. W. and Whitt, W. (1989). Indirect estimation via l= λw. *Operations Research*, 37(1):82–103.

Hammersley, J. M. and Morton, K. W. (1956). A new Monte Carlo technique: antithetic variates. *Mathematical proceedings of the Cambridge philosophical society*, 52(3):449–475.

Hesterberg, T. C. and Nelson, B. L. (1998). Control variates for probability and quantile estimation. *Management Science*, 44(9):1295–1312.

Kahn, H. and Marshall, A. (1953). Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278.

Karr, A. F. (1993). *Probability*. Springer, New York.

L'Ecuyer, P. (1994). Efficiency improvement and variance reduction. In *Proceedings of the 1994 Winter Simulation Conference*, pages 122–132.

L'Ecuyer, P., Simard, R., Chen, E. J., and Kelton, W. D. (2002). An object-oriented random number package with many long streams and substreams. *Operations research*, 50(6):131–137.

Luenberger, D. G. (1998). *Investment Science*. Oxford University Press, New York.

Lunney, P. D. and Anderson, C. A. (2009). Investigation of the statistical power of the content uniformity tests using simulation studies. *Journal of Pharmaceutical Innovation*, 4(1):24–35.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM transactions on modeling and computer simulation*, 8(1):3–30.

Owen, A. B. (2001). *Empirical Likelihood*. Chapman & Hall/CRC, Boca Raton, FL.

Pullin, D. I. (1979). Generation of normal variates with given sample mean and variance. *Journal of Statistical Computation and Simulation*, 9(4):303–309.

Wilson, A. (1965). *The Casino Gambler's Guide*. Harper & Row, New York.

Wilson, J. R. (1984). Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences*, 4(3):277–312.