Contents

10

Advanced variance reduction

This chapter collects together some advanced and specialized topics in variance reduction. They are generalizations, extensions and hybrids of methods previously considered.

We begin with §10.1 on grid-based stratification, suitable for low dimensions In dimension d it improves the Monte Carlo RMSE to $O(n^{-1/2-1/d})$. In §10.2 we apply antithetic sampling within those strata, yielding a method of Haber with RMSE $O(n^{-1/2-2/d})$. Then §10.3 presents Latin hypercube sampling, a stratification method suitable for large or even unbounded dimension. We round out our mini-chapter on advanced stratification with §10.4 on orthogonal array sampling, which is very well suited to intermediate dimensionalities.

Importance sampling can provide great efficiency gains, but it is difficult to do well and a poor choice can severely reduce efficiency. Adaptive importance sampling (§10.5) uses importance sampled data to choose a new density for further importance sampling. We study nonparametric versions of adaptive importance sampling in §10.6.

Generalizations of antithetic sampling (§10.7) use clusters of two or more points to reduce variance. We look at combining control variates with antithetic or stratified sampling in §10.8. One very advanced use of importance sampling is to estimate normalizing constants or ratios of them as described in §10.9 on bridge, umbrella and path sampling, as well as warp bridge sampling.

10.1 Grid-based stratification

For stratification to bring an enormous efficiency gain, we need our integrand f to be nearly constant within strata. That is, f must be highly correlated

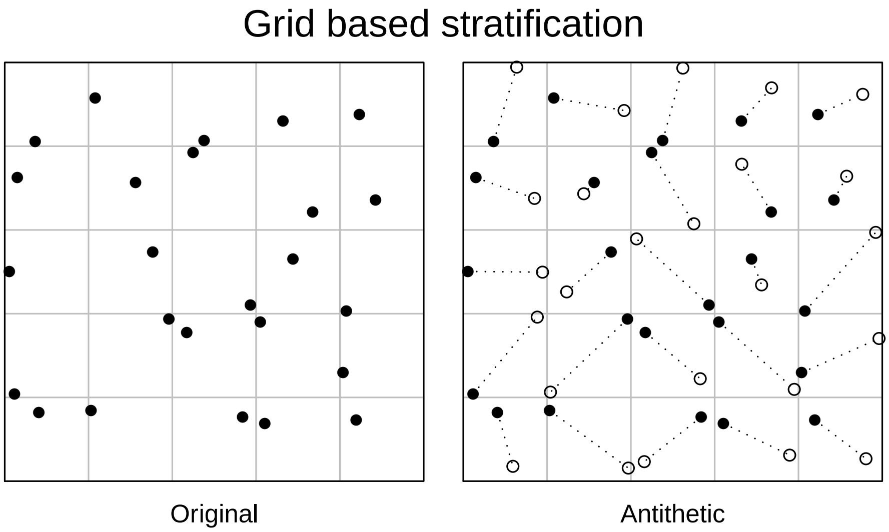

Figure 10.1: The left panel shows an unreplicated grid-based stratified sample of 25 points in $[0,1)^2$. The right panel shows the combination with antithetic sampling described in §10.2.

with a piece-wise constant function. Such high correlation can be hard to arrange with a small number of strata, but if we allow the number of strata to grow proportionally to n we can get much more close-fitting piece-wise constant functions.

We will work with $\mu = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, where $\boldsymbol{X} \sim \mathbf{U}[0,1)^d$ may represent the uniform random variables that we transform into the random quantities the problem uses. The d dimensional unit cube $[0,1)^d$ can be split into m^d congruent subcubes with sides of length $1/m$. Let $\boldsymbol{X}_i$ be sampled uniformly within the i'th cube (listed in any order) for $i = 1,\dots,n = m^d$. The left panel of Figure 10.1 illustrates this sample.

The grid-based stratified estimate is now

$$\hat{\mu}_{\mathrm{gs}} = \frac{1}{m^d}\sum_{i=1}^{m^d} f(\boldsymbol{X}_i)$$

where $\boldsymbol{X}_i$ is uniformly distributed within a cube of volume m^{-d}, whose center we will call $\boldsymbol{c}_i$. The uniformity of the $\boldsymbol{X}_i$ imply that $\mathbb{E}(\hat{\mu}_{\mathrm{gs}}) = \mu$. We need only consider the variance of $\hat{\mu}_{\mathrm{gs}}$.

Now, suppose that f is continuously differentiable on $[0,1]^d$. It follows that the first derivative of f is uniformly continuous on $[0,1]^d$ and hence also on the integration region $[0,1)^d$ (which partitions more cleanly than $[0,1]^d$ does). Then

$$f(\boldsymbol{X}_i) = L_i + Z_i, \quad \text{where} \tag{10.1}$$

$$
\begin{aligned}
L_i &= f(\boldsymbol{c}_i) + \sum_{j=1}^{d} \frac{\partial}{\partial x_j} f(\boldsymbol{c}_i)(X_{ij} - c_{ij}), \quad \text{and} \\
Z_i &= o\Bigl(\frac{1}{m}\Bigr).
\end{aligned}
$$

Equation (10.1) has a linear term L_i and an error term Z_i. We consider first the linear term:

$$
\mathrm{Var}(L_i) = \frac{1}{12m^2} \sum_{j=1}^{d} \Bigl(\frac{\partial}{\partial x_j} f(\boldsymbol{c}_i)\Bigr)^2 = \frac{1}{12m^2} \|\nabla f(\boldsymbol{c}_i)\|^2,
$$

because $X_{ij} - c_{ij}$ are independent $\mathbf{U}[-1/(2m), 1/(2m))$ random variables which have variance $1/(12m^2)$. Now, the variance of $\sum_{i=1}^{n} L_i$ is

$$
\begin{aligned}
\frac{1}{n^2 12m^2} \sum_{i=1}^{n} \|\nabla f(\boldsymbol{c}_i)\|^2 &\to \frac{1}{n12m^2} \int \|\nabla f(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{12n^{1+2/d}} \int \|\nabla f(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x}
\end{aligned}
$$

as m and n go to ∞.

As for the error terms Z_i, each of them has variance $o(1/m^2)$ and their average has variance $o(1/(nm^2))$ which is negligible compared to the variance of the average of L_i. Similarly, the covariance of $(1/n)\sum_i L_i$ and $(1/n)\sum_i Z_i$ is negligible compared to the variance of $(1/n)\sum_i L_i$. As a result

$$
\lim_{n\to\infty} n^{1+2/d}\,\mathrm{Var}(\hat{\mu}_{\mathrm{gs}}) = \frac{1}{12} \int \|\nabla f(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x}, \tag{10.2}
$$

the limit being taken through values $n = m^d$ as $m \to \infty$.

What is striking about (10.2) is that the error variance does not decrease at the usual $1/n$ rate. It is $O(n^{-1-2/d})$ instead. For small dimensions d, the improvement in (10.2) compared to the usual variance σ^2/n, can be dramatic. For example, when $d = 1$ the stratification variance is $O(n^{-3})$ compared to $O(n^{-1})$ for ordinary Monte Carlo. It is comparable to cubing the sample size. For $d = 2$ the variance is $O(n^{-2})$. For larger d, the variance decrease from stratification becomes small. Even though the factor $n^{2/d}$ tends to infinity with n, plugging in realistic sample sizes n and a large dimension d, gives a meager result.

The most direct way to estimate the variance of $\hat{\mu}_{\mathrm{gs}}$ is to take $k \geqslant 2$ replicates within each subcube. The usual choice is $k = 2$ because that allows the largest value of m for $n = km^d$ function evaluations. Let $\boldsymbol{X}_i$ and $\boldsymbol{X}_i'$ be independently sampled uniformly within the subcube centered at $\boldsymbol{c}_i$. Then we may use

$$
\begin{aligned}
\hat{\mu}_{\mathrm{gs}} &= \frac{1}{2m^d} \sum_{i=1}^{m^d} (f(\boldsymbol{X}_i) + f(\boldsymbol{X}_i')), \quad \text{and} \\
\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{gs}}) &= \frac{1}{4m^{2d}} \sum_{i=1}^{m^d} (f(\boldsymbol{X}_i) - f(\boldsymbol{X}_i'))^2.
\end{aligned} \tag{10.3}
$$

For very large d, stratification becomes cumbersome to implement. The smallest available sample size is 2^d which may already be too large to use. Even if 2^d is not too large, there is no good way to extend a grid-based stratified sample of m^d points to one with $(m+1)^d$ points. We could take samples of size km^d for integer $k \geqslant 2$ by replicating the stratification but the variance of $\hat{\mu}_{\mathrm{gs}}$ along a sequence with increasing k and fixed m is the ordinary $O(n^{-1})$ rate.

Stratifying works so well for small d, because the function f becomes nearly constant within cells of side $1/m$. In §10.2 a combination of stratification and antithetic sampling exploits near linearity of f within such cells to reduce the variance to $O(n^{-1-4/d})$.

In practice, there is no need to use the same value of m on every dimension, and it is straightforward to use an $m_1 \times m_2 \times \cdots \times m_d$ grid of strata. Also, to allow variance estimation, we should take 2 observations per cell.

The practical utility of fully d-dimensional grid-based stratification is limited. It brings a large improvement over Monte Carlo when d is small and f is smooth. But that case is already well handled by classical quadrature methods, discussed in Chapter 7. Compared to classical methods, stratification has the advantage of providing unbiased estimates along with variance estimates. But we can usually expect better results from randomized quasi-Monte Carlo (Chapter 17). It remains useful though to stratify on a small subset of the most important input variables. This strategy will usually be outperformed by randomized quasi-Monte Carlo, but it has the advantage of a simpler implementation.

10.2 Stratification and antithetics

Antithetic sampling can be combined with stratification. A natural way to do this is to apply antithetic sampling within some, typically all, of the strata. Let there be strata $\mathcal{D}_j$ for $j = 1, \dots, J$ with $\omega_j = \mathbb{P}(\boldsymbol{X} \in \mathcal{D}_j)$ and within-stratum distributions $p_j(\boldsymbol{x}) = p(\boldsymbol{x})1_{\boldsymbol{x}\in\mathcal{D}_j}/\omega_j$.

We take $\boldsymbol{X}_{ij} \sim p_j$ for $i = 1, \dots, n_j$ and $j = 1, \dots, J$. Then we take antithetic counterparts $\widetilde{\boldsymbol{X}}_{ij}$ appropriate to their strata. If p_j is symmetric about $\boldsymbol{c}_j \in \mathcal{D}_j$ then we may take $\widetilde{\boldsymbol{X}}_{ij} = 2\boldsymbol{c}_j - \boldsymbol{X}_{ij}$. We estimate $\mu_j = \int f(\boldsymbol{x})p_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by

$$\hat{\mu}_{\mathrm{anti},j} = \frac{1}{2n_j}\sum_{i=1}^{n_j} f(\boldsymbol{X}_{ij}) + f(\widetilde{\boldsymbol{X}}_{ij}) = \frac{1}{n_j}\sum_{i=1}^{n_j} f_{j,\mathrm{E}}(\boldsymbol{X}_{ij}),$$

where $f_{j,\mathrm{E}} = (f(\boldsymbol{X}_{ij}) + f(\widetilde{\boldsymbol{X}}_{ij}))/2$.

The combined estimate is

$$\hat{\mu}_{\mathrm{strat,anti}} = \sum_{j=1}^{J} \omega_j \hat{\mu}_{\mathrm{anti},j}. \tag{10.4}$$

If all $n_j \geqslant 2$, then we have a variance estimate

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\mathrm{strat,anti}}) = \sum_{j=1}^{J} \frac{\omega_j^2}{n_j - 1} \sum_{i=1}^{n_j} f_{j,\mathrm{E}}(\boldsymbol{X}_{ij} - \hat{\mu}_{\mathrm{anti},j})^2. \tag{10.5}$$

Antithetic sampling within each of cell of an m^d grid can be extremely effective when f is smooth. The error of $\hat{\mu}_{\mathrm{gs}}$ for stratified sampling within subcubes, from §10.1, was dominated by a linear term within each subcube. Antithetic sampling integrates those dominant terms with zero error.

We split $[0,1)^d$ into m^d congruent subcubes with side $1/m$. We will assume that f has a uniformly bounded second derivative matrix on $[0,1)^d$, as it would have if its second derivative were continuous on the closed cube $[0,1]^d$.

Let $\boldsymbol{X}_i$ be sampled uniformly within the cubical cell centered at $\boldsymbol{c}_i$ and put $\widetilde{\boldsymbol{X}}_i = 2\boldsymbol{c}_i - \boldsymbol{X}_i$. The right panel of Figure 10.1 illustrates this sample. With this strategy, we have $n = 2m^d$ sample points and the estimate of μ is

$$\hat{\mu}_{\mathrm{gs,anti}} = \frac{1}{2m^d} \sum_{i=1}^{n} \bigl(f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i)\bigr) = \frac{1}{m^d} \sum_{i=1}^{n} f_{i,\mathrm{E}}(\boldsymbol{X}_i).$$

The expected value of $f_{i,\mathrm{E}}(\boldsymbol{X}_i)$ is the integral of f over the subcube containing $\boldsymbol{c}_i$. As a result $\mathbb{E}(\hat{\mu}_{\mathrm{gs,anti}}) = \mu$.

It remains to study the variance of $\hat{\mu}_{\mathrm{gs,anti}}$. The function $f_{i,\mathrm{E}}$ has first order partial derivatives equal to 0 at $\boldsymbol{c}_i$ by symmetry. It has uniformly bounded second order derivatives because f does. Finally, the distance from $\boldsymbol{X}_i$ to $\boldsymbol{c}_i$ is at most $\sqrt{d}/(2m)$ because $\boldsymbol{X}_i$ is in a d dimensional cube of side $1/m$ centered at $\boldsymbol{c}_i$. Putting these together we have

$$f_{i,\mathrm{E}}(\boldsymbol{X}_i) = f_{i,\mathrm{E}}(\boldsymbol{c}_i) + O(m^{-2}) = f(\boldsymbol{c}_i) + O(m^{-2})$$

with an implied constant that is uniform in $\boldsymbol{X}_i$ and in n. Averaging over $\boldsymbol{X}_i$ yields $\mu_i = f(\boldsymbol{c}_i) + O(m^{-2})$ too. Therefore

$$\begin{aligned} \mathrm{Var}(\hat{\mu}_{\mathrm{gs,anti}}) &= \frac{1}{n^2} \sum_{i=1}^{n} \mathbb{E}((f_{i,\mathrm{E}}(\boldsymbol{X}_i) - \mu_i)^2) \\ &= O(n^{-1}m^{-4}) = O(n^{-1-4/d}). \end{aligned}$$

If we want a variance estimate then we can take 2 or more antithetic pairs inside every subcube and use equation (10.5).

As for the grid-based stratification of §10.1, the value of the higher rate of convergence diminishes with dimension and the method is awkward to apply in very high dimensions. The combination of antithetic sampling within cubic strata remains valuable for use on a small number of very important input variables.

There is one minor theoretical complication. If $\boldsymbol{X}_i$ is on the lower boundary of its cell, then $\widetilde{\boldsymbol{X}}_i$ is on the corresponding upper boundary. The cells are

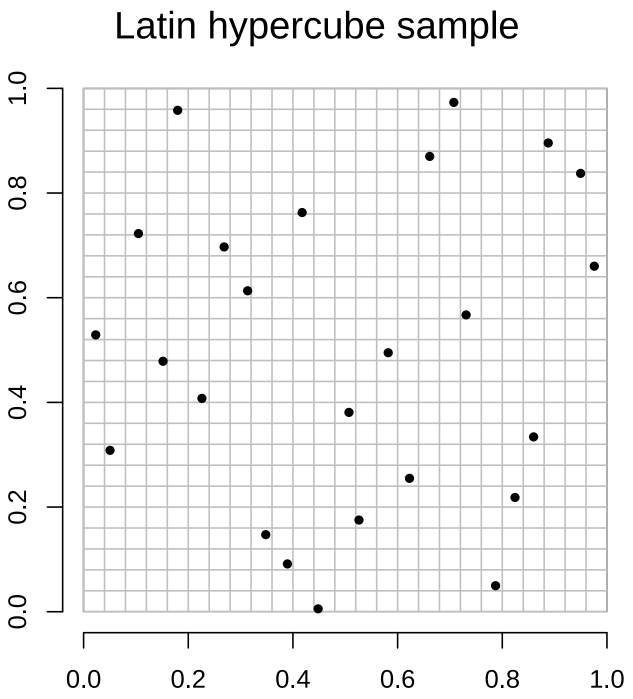

Figure 10.2: This figure shows 25 points of a Latin hypercube sample in $[0,1]^2$.

constructed to contain their lower, but not upper boundaries. As a result $\widetilde{\boldsymbol{X}}_i$ is not always strictly inside its own stratum. In the extreme it is possible for $\widetilde{\boldsymbol{X}}_i$ to be outside of $[0,1)^d$, but of course, inside $[0,1]^d$. These events have zero probability of arising. They don't affect the sampling algorithm, but do add a complication to its description.

10.3 Latin hypercube sampling

When one of the d components of $\boldsymbol{X}$ is particularly important, we can improve on Monte Carlo by stratifying that component. If we want to stratify on all of $d > 1$ components, then we may use grid-based stratification of §10.1. But a regular grid only has $n^{1/d} \ll n$ strata per component. Also for large d, grids are infeasible.

For higher accuracy, we want to use more levels than grid-based stratification will allow. In ***Latin hypercube sampling*** (LHS) we generate a sample in which each of d components for $\boldsymbol{X} \sim \mathbf{U}(0,1)^d$ is stratified into n equal strata. That is Figure 10.2 illustrates a Latin hypercube sample of $n = 25$ points in $d = 2$ dimensions. A grid of lines is overlaid. Each of 25 horizontal strata gets one point. The same is true for each of 25 vertical strata.

There is a simple formula for Latin hypercube sampling:

$$X_{ij} = \frac{\pi_j(i-1)+U_{ij}}{n}, \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant d \tag{10.6}$$

where $\pi_1, \dots, \pi_d$ are uniform random permutations of $\{0, 1, \dots, n-1\}$, $U_{ij} \sim \mathbf{U}[0,1)$, and all the U_{ij} and π_j are independent.

Latin hypercube sampling has a multiple stratification property: for all $a = 0, \dots, n-1$ and all $j = 1, \dots, d$ we have

$$\#\biggl\{1 \leqslant i \leqslant n \mid \frac{a}{n} \leqslant X_{ij} < \frac{a+1}{n}\biggr\} = 1. \tag{10.7}$$

With just n points we balance nd strata of volume $1/n$ each. We can even use LHS with $d > n$.

Latin hypercube sampling has been discovered several times in different literatures. It is known as ***n*-rooks** in computer graphics. If we interpret the 25 points in Figure 10.2 as rooks on a 25 × 25 chessboard, none of them could capture any of the others. A centered version given in (10.8) below was called ***lattice sampling*** in an agricultural context. See page 52 of the chapter end notes.

The multiple stratification property (10.7) follows easily from equation (10.6). Let I_a be the interval $[a/n, (a+1)/n)$ for $a \in \{0, \dots, n-1\}$. Now $X_{ij} \in I_a$ if and only if $\pi_j(i-1) = a$. This in turn happens if and only if $i-1 = \pi_j^{-1}(a)$. For each $j = 1, \dots, d$ there is precisely one such i. Therefore LHS satisfies (10.7).

LHS combines stratification with uniform distribution. Theorem 10.1 shows that each of the LHS points $\boldsymbol{X}_i$ has the $\mathbf{U}[0,1)^d$ distribution.

Theorem 10.1. *Let $d \geqslant 1$ and $n \geqslant 1$ be integers. Let X_{ij} be a Latin hypercube sample defined by* (10.6)*. Then $\boldsymbol{X}_i \sim \mathbf{U}[0,1)^d$ holds for each $i = 1, \dots, n$.*

Proof. First $\boldsymbol{X}_i = (X_{i1}, \dots, X_{id})$ has independent components. So it is enough to show that each $X_{ij} \sim \mathbf{U}[0,1)$. Let $A = \pi_j(i-1)$. Then $A \sim \mathbf{U}\{0, 1, \dots, n-1\}$. The conditional distribution of X_{ij} given that $A = a$ is $\mathbf{U}[a/n, (a+1)/n)$ which has probability density function

$$f_a(x) = \begin{cases} n, & a/n \leqslant x < (a+1)/n \\ 0, & \text{else.} \end{cases}$$

The probability density function of X_{ij} is $(1/n)\sum_{a=0}^{n-1} f_a(x)$ which equals 1 if $0 \leqslant x < 1$ and 0 otherwise. □

The LHS estimate of $\mu = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is simply

$$\hat\mu_{\mathrm{LHS}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{X}_i)$$

with $\boldsymbol{X}_i$ generated by (10.6). An immediate consequence of Theorem 10.1 is that $\mathbb{E}(\hat\mu_{\mathrm{LHS}}) = \mu$.

An alternative form that is sometimes useful takes

$$X_{ij} = \frac{\pi_j(i-1)+1/2}{n}, \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant d. \tag{10.8}$$

We write $\hat\mu_{\mathrm{LHS}-c} = (1/n)\sum_{i=1}^n f(\boldsymbol{X}_i)$ for this centered version of LHS. The centered LHS has a small bias of $O(n^{-2})$.

LHS can be much more accurate than plain Monte Carlo. Intuitively we would expect LHS to perform well whenever stratification on one of the d components x_j would be effective. We would not even need to know in advance which $j \in \{1,\dots,d\}$ was the important one. It turns out the LHS works well in somewhat greater generality.

The favorable circumstance for LHS arises when $f(\boldsymbol{x})$ is nearly an additive function of the d components of $\boldsymbol{x}$. It is convenient to write the additive function as $f_0 + \sum_{j=1}^d f_j(x_j)$ where $\int_0^1 f_j(x)\,\mathrm{d}x = 0$ and $f_0 \in \mathbb{R}$. Centering the f_j to have mean 0 removes an indeterminacy in which we could have replaced f_j by $f_j + c$ while simultaneously replacing f_k by $f_k - c$ for $k \neq j$ and $c \in \mathbb{R}$.

Using the ANOVA decomposition of the unit cube $[0,1]^d$ (see Appendix A) we may identify the best additive approximation to f. It is

$$\begin{aligned} f^{\mathrm{add}}(\boldsymbol{x}) &= \mu + \sum_{j=1}^d f_j(x_j), \quad \text{where} \\ \mu &= \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad \text{and} \\ f_j(x_j) &= \int_{[0,1]^{d-1}} (f(\boldsymbol{x})-\mu) \prod_{k\in\{1,\dots,d\}-\{j\}} \mathrm{d}x_k. \end{aligned}$$

The function f^{add} is the best additive approximation to f in the following sense: If $g(\boldsymbol{x})$ is any other additive function on $[0,1]^d$ then, by Lemma A.6,

$$\int (g(\boldsymbol{x}) - f(\boldsymbol{x}))^2\,\mathrm{d}\boldsymbol{x} \geqslant \int (f^{\mathrm{add}}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\,\mathrm{d}\boldsymbol{x}.$$

Now define the residual from additivity $e(\cdot)$ by $f(\boldsymbol{x}) = f^{\mathrm{add}}(\boldsymbol{x}) + e(\boldsymbol{x})$. Then, in the centered version (10.8):

$$\begin{aligned} \hat\mu_{\mathrm{LHS}-c} &= \frac{1}{n}\sum_{i=1}^n \Bigl(\mu + \sum_{j=1}^d f_j(X_{ij}) + e(\boldsymbol{X}_i)\Bigr) \\ &= \mu + \sum_{j=1}^d \frac{1}{n}\sum_{i=1}^n f_j\Bigl(\frac{i-1/2}{n}\Bigr) + \frac{1}{n}\sum_{i=1}^n e(\boldsymbol{X}_i), \end{aligned}$$

after reordering the points X_{ij} for each fixed j.

We see that the LHS error $\hat\mu_{\mathrm{LHS}-c} - \mu$ is a summed error of d midpoint rules applied to f_j, plus the LHS estimate of the mean of $e(\boldsymbol{x})$. When the f_j are smooth enough, then the midpoint rule is much more accurate than plain Monte

Carlo for estimating $\int_0^1 f_j(x)\,\mathrm{d}x$. The average of $e(\boldsymbol{X}_i)$ is typically $O_p(1/\sqrt{n})$ and then it dominates the error of LHS. So if f is nearly additive, in the sense that $e(\boldsymbol{X})$ has small variance, then LHS is a big improvement over ordinary Monte Carlo. For the unbiased version (10.6), LHS uses an estimate that has stratified sampling instead of midpoint rules for the g_j. Once again the error is dominated by the average of $e(\boldsymbol{X}_i)$.

Next we list some facts about Latin hypercube sampling.

Proposition 10.1. *Let $f(\boldsymbol{x})$ be a real-valued function on $[0,1]^d$ with $\int f(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} < \infty$. Let $\boldsymbol{X}_1,\dots,\boldsymbol{X}_n$ be the unbiased Latin hypercube sample defined by* (10.6) *and let $\hat\mu_{\mathrm{LHS}} = (1/n)\sum_{i=1}^n f(\boldsymbol{X}_i)$. Then*

$$\mathrm{Var}(\hat\mu_{\mathrm{LHS}}) = \frac{1}{n}\int e(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} + o\Bigl(\frac{1}{n}\Bigr),$$

where $e(\boldsymbol{x}) = f(\boldsymbol{x}) - f^{\mathrm{add}}(\boldsymbol{x})$.

Proof. Stein (1987). □

Proposition 10.1 shows us that the additive part of f does not contribute to the asymptotic variance in Latin hypercube sampling. It is as if the additive part were simply not there, at least for large n. The normalized error $\sqrt{n}(\hat\mu_{\mathrm{LHS}} - \mu)$ has the same mean and asymptotically the same variance as $\sqrt{n}$ times $(1/n)\sum_{i=1}^n e(\boldsymbol{X}_i)$.

Proposition 10.2. *Let $f(\boldsymbol{x})$ be a bounded function on $[0,1]^d$ and put $e(\boldsymbol{x}) = f(\boldsymbol{x}) - f^{\mathrm{add}}(\boldsymbol{x})$. Let $\bar e_{\mathrm{LHS}} = (1/n)\sum_{i=1}^n e(\boldsymbol{X}_i)$ where $\boldsymbol{X}_1,\dots,\boldsymbol{X}_n$ are points of a Latin hypercube sample defined by* (10.6). *Let $\widetilde{\boldsymbol{X}}_1,\dots,\widetilde{\boldsymbol{X}}_n$ be independent $\mathbf{U}[0,1]^d$ random vectors and set $\bar e_{\mathrm{IID}} = (1/n)\sum_{i=1}^n e(\widetilde{\boldsymbol{X}}_i)$. Then for integer $p \geqslant 1$*

$$\mathbb{E}\bigl((\sqrt{n}\,\bar e_{\mathrm{LHS}})^p\bigr) = \mathbb{E}\bigl((\sqrt{n}\,\bar e_{\mathrm{IID}})^p\bigr) + O(n^{-1})$$

as $n \to \infty$.

Proof. Lemma 2 of Owen (1992a). □

Not only does $\sqrt{n}\,\bar e_{\mathrm{IID}}$ have the same mean and variance as $\sqrt{n}\,\bar e_{\mathrm{LHS}}$ but higher moments converge too. So a good heuristic is that $\hat\mu$ behaves like μ plus the average of n IID $e(\boldsymbol{X}_i)$ random variables. As a result a central limit theorem holds for Latin hypercube sampling of bounded random variables. The central limit theorem holds given only finite third moments:

Proposition 10.3. *Let $f(\boldsymbol{x})$ be a function from $[0,1]^d$ to $\mathbb{R}^k$ for $1 \leqslant k \leqslant \infty$. Suppose that $\int \|f(\boldsymbol{x})\|^3\,\mathrm{d}\boldsymbol{x} < \infty$ and $\Sigma = \int (f(\boldsymbol{x}) - f^{\mathrm{add}}(\boldsymbol{x}))(f(\boldsymbol{x}) - f^{\mathrm{add}}(\boldsymbol{x}))^{\mathsf{T}}\,\mathrm{d}\boldsymbol{x}$ has rank k. Then $\sqrt{n}(\hat\mu_{\mathrm{LHS}} - \mu) \xrightarrow{\mathrm{d}} \mathcal{N}(0,\Sigma)$ as $n \to \infty$.*

Proof. Corollary 2 of Loh (1996). □

x	y	x+y	x+2y
0	0	0	0
0	1	1	1
0	2	2	2
1	0	1	2
1	1	2	0
1	2	0	1
2	0	2	1
2	1	0	2
2	2	1	0

Table 10.1: An orthogonal array of type $\mathrm{OA}(9, 4, 3, 2)$.

Proposition 10.4. *Let $f(\boldsymbol{x})$ be a real-valued function on $[0,1]^d$ with $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and $\sigma^2 = \int (f(\boldsymbol{x}) - \mu)^2\,\mathrm{d}\boldsymbol{x} < \infty$. For $n \geqslant 2$ let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ be a Latin hypercube sample. Then* $\mathrm{Var}(\hat\mu_{\mathrm{LHS}}) \leqslant \sigma^2/(n-1)$.

Proof. Owen (1997). □

The point of Proposition 10.4 is that while Latin hypercube sampling can be much better than IID sampling, it cannot be much worse. At the worst, LHS has a variance $n/(n-1)$ times as large as under IID sampling on n values, as if we had thrown out one sample point.

LHS is easy to implement, can greatly reduce variance for nearly additive functions, and otherwise does not make the variance much larger. It is somewhat more difficult to estimate the variance of $\hat\mu_{\mathrm{LHS}}$ than it is for IID sampling. The most straightforward way is to use some independent replicates of the Latin hypercube sample.

10.4 Orthogonal array sampling

Latin hypercube sampling stratifies the components of $\boldsymbol{X}_i$ one at a time. We might be able to do better by stratifying all pairs of components. The combinatorial object that does this is known as an orthogonal array. By randomizing an orthogonal array and embedding it into the unit cube $[0,1]^d$ we obtain a generalization of Latin hypercube sampling.

Definition 10.1. For integers $b \geqslant 2$ and $d \geqslant t \geqslant 1$, let $A \in \{0, 1, \dots, b-1\}^{n\times d}$. The matrix A is called an ***orthogonal array of strength t*** if each n by t submatrix of A contains all b^t possible distinct rows, the same number λ of times. We denote such a matrix by $\mathrm{OA}(n, d, b, t)$.

An example orthogonal array is shown in in Table 10.1. There are 4 columns. Each 9 by 2 submatrix has all 9 values from $\{0, 1, 2\}^2$.

The d columns of A are called constraints. Orthogonal arrays are often defined with n columns and d rows instead, which makes the smaller ones easier to print. The arrangement here has a row for each sample point we will generate and a column for each component that we will sample.

If we were to plot an n by t submatrix of A in $\mathbb{R}^t$ using one data point for each row of the submatrix in the natural way, then the result would be a b^t grid of points each one repeated λ times.

Counting the rows shows that $n = \lambda b^t$. The number λ is called the index of the array. There are good orthogonal arrays with $\lambda = 1$.

Example 10.1 (Bose construction)**.** The Bose construction gives us orthogonal arrays of type $\mathrm{OA}(p^2, p+1, p, 2)$ where p is a prime number. In the Bose construction $A_{i1} = \lfloor (i-1)/p \rfloor$ and $A_{i2} = (i-1) \bmod p$ for $i = 1, \dots, p^2$. The first two columns of A go through their p^2 possibilities just like the last two digits in a base p odometer. See the first two columns of Table 10.1 for an example. To complete the array, put $A_{ij} = A_{i1} + (j-2)A_{i2} \bmod p$ for $3 \leqslant j \leqslant p+1$. If we name the first and second columns x and y respectively, then the array also contains columns $x + ay$ for $a = 1, \dots, p-1$.

The Bose construction is easy to apply. It requires a prime base p. If for example, one carries out the algorithm in base 4 or 6 the result is certainly not an orthogonal array of strength 2. The Bose construction can be generalized to $\mathrm{OA}(b^2, b+1, b, 2)$ where $b = p^r$ for prime p and any integer $r \geqslant 1$. The generalization requires arithmetic in the finite field of p^r elements. Using modular arithmetic will not be correct for $r > 1$.

To understand why the Bose construction works, pick two values a_1 and a_2, not necessarily distinct, from $\{0, 1, \dots, p-1\}$. Now pick two distinct columns from A, indexed for example by j_1 and j_2 where $1 \leqslant j_1 < j_2 \leqslant p+1$. If we can always find a unique row i where $A_{ij_1} = a_1$ and $A_{ij_2} = a_2$ both hold, then A is a strength 2 array as required.

It is convenient to index the row i by the values of A_{i1} and A_{i2} respectively. For simplicity we'll call these values x and y respectively, which belong to $\{0, 1, \dots, p-1\}$. If neither j_1 nor j_2 is 2 then the values of x and y satisfy

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 & c_1 \\ 1 & c_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

for distinct c_1 and c_2 in $\{0, \dots, p-1\}$. The solution is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & c_1 \\ 1 & c_2 \end{pmatrix}^{-1} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

with all the arithmetic done modulo p. Integers modulo a prime p form a finite field, and so the arithmetic is well defined unless the matrix is singular. Then its inversion would require division by 0. But the matrix has determinant $c_2 - c_1$ which is not zero. Therefore

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{c_2 - c_1} \begin{pmatrix} c_2 & -c_1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} (a_1 c_2 - c_1 a_2)/(c_2 - c_1) \\ (a_2 - a_1)/(c_2 - c_1) \end{pmatrix}$$

exists (and is unique).

If one of the columns that we chose (say j_1) was column 2 of A, then the critical matrix in the above analysis becomes $\left(\begin{smallmatrix}0 & 1\\ 1 & c\end{smallmatrix}\right)$ for $c \in \{0, \dots, p-1\}$. This matrix is also nonsingular, having determinant $-1 \equiv p - 1 \mathsf{~mod~} p$. So in every case we can find the required x and y, hence the required row i, and so A is indeed $\mathrm{OA}(p^2, p+1, p, 2)$.

The Bose construction gives $p^3(p+1)/2$ properly balanced strata of volume $1/n$ using $n = p^2$ sample values. That is, the number of strata is larger than $n^2/2$ for a sample size n. When $p = 101$ the Bose array constructs $n = 10201$ points in up to 102 dimensions. All $\binom{102}{2} = 102 \times 101/2 = 5151$ pairs of columns of the array plot as regular 101 by 101 grids. The design balances over 5×10^7 strata.

The Bose construction requires p^2 points where $p \geqslant d - 1$, so for very large d it may be unwieldy. For small d we only need to construct the first $d \leqslant p + 1$ columns of the array.

To use the Bose construction in a Monte Carlo context, we could take $x_{ij} = A_{ij}/p$. But this direct embedding into $[0,1]^p$ has a serious flaw. Consider the first three columns, named x, y and $x + y \mathsf{~mod~} p$. The values of (A_{i1}, A_{i2}, A_{i3}) lie within just two planes in $[0, p-1]^3$. Other triples in the Bose construction are also flawed, lying within just a small number of planes, but the first three columns are most seriously affected.

Randomization serves to break up the planar flaw in the Bose construction. A ***randomized orthogonal array*** sample has

$$X_{ij} = \frac{\pi_j(A_{ij}) + U_{ij}}{b}, \quad \text{for } 1 \leqslant i \leqslant n \quad \text{and } 1 \leqslant j \leqslant d, \tag{10.9}$$

where A is an orthogonal array of type $\mathrm{OA}(n, d, b, t)$, $\pi_1, \dots, \pi_d$ are uniform random permutations of $\{0, \dots, b-1\}$, each $U_{ij} \sim \mathbf{U}[0,1)$ and the random components π_j and U_{ij} are all mutually independent. The centered version takes

$$X_{ij} = \frac{\pi_j(A_{ij}) + 1/2}{b} \in [0,1)^d \tag{10.10}$$

under the same conditions on A and π_j. Figure 10.3 illustrates an orthogonal array A_{ij} and the values $\pi_j(A_{ij})$ for its randomized counterpart. Randomizing the levels of the orthogonal array is essential.

If $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ are defined via (10.9) then $\boldsymbol{X}_i \sim \mathbf{U}[0,1)^d$ by the same arguments that served for Latin hypercube sampling in Theorem 10.1.

Randomized orthogonal array sampling has a t-dimensional stratification property which generalizes the one for LHS at (10.7). Let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ be a randomized orthogonal array of type $\mathrm{OA}(n, d, b, t)$. Then for every $\boldsymbol{a} \in \{0, 1, \dots, b-1\}^t$ and every $u \subset \{1, \dots, d\}$ with $|u| = t$,

$$\#\Bigl\{1 \leqslant i \leqslant n \mid \frac{a_j}{b} \leqslant X_{ij} < \frac{a_j + 1}{b}, \quad \forall j \in u\Bigr\} = nb^{-t} = \lambda. \tag{10.11}$$

Randomized orthogonal array sampling generalizes Latin hypercube sampling. The LHS corresponds to use of a strength $t = 1$ array with $A_{ij} = i$

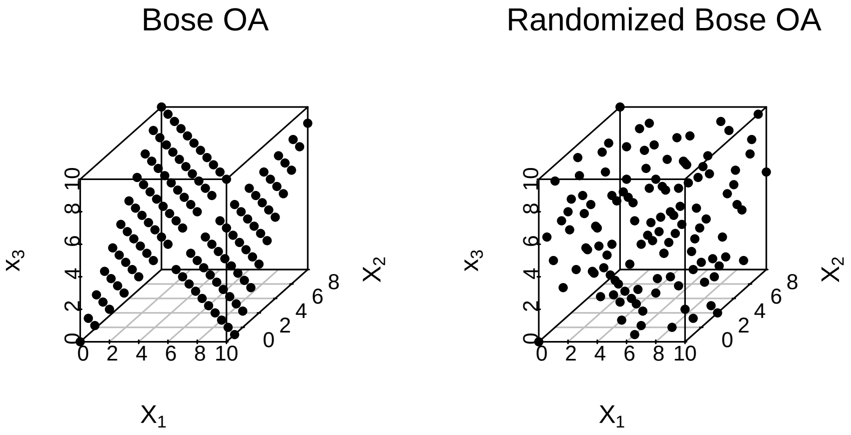

Figure 10.3: This figure shows 121 points of a Bose orthogonal array $\mathrm{OA}(121, 3, 11, 2)$ and the same points randomized.

for $i = 1, \dots, n$ and $j = 1, \dots, d$. Full grid-based stratification of $[0, 1]^d$ also corresponds to randomized orthogonal array sampling, using strength $t = d$.

In Latin hypercube sampling the sample mean $\hat{\mu}$ had a variance close to $(1/n) \sum_{|u|>1} \sigma_u^2$ where σ_u^2 are the variances from the ANOVA decomposition of f. For randomized orthogonal array sampling we expect a variance something like $(1/n) \sum_{|u|>t} \sigma_u^2$ when $t < d$. See references on page 52 of the end notes. For $t = d$ the variance is of smaller order than $1/n$. These variance results all assume that f is twice continuously differentiable.

Theorem 10.2. *Let $b = p^r$ where p is a prime number and r is a positive integer. Then for an integer $t \geqslant 1$ with $t < b$ the orthogonal array $OA(b^t, b + 1, b, t)$ exists.*

Proof. Bush (1952). □

Example 10.2 (Bush construction)**.** The Bose construction was generalized by Bush. Here we give Bush's construction of $\mathrm{OA}(p^t, p+1, p, t)$ for $1 \leqslant t < p$ where p is a prime number. In this case we can use arithmetic modulo p. The general case with $b = p^r$ for $r > 1$ requires Galois field arithmetic (and arithmetic modulo b is then incorrect).

The construction uses p^t distinct polynomials of the form

$$\phi_i(x) = a_{i,t-1}x^{t-1} + \cdots + a_{i,1}x + a_{i,0}$$

for $0 \leqslant i < p^t$, where $a_{i,0}, \dots, a_{i,t-1} \in \{0, \dots, p-1\}$. For sake of definiteness we can pick the coefficients so that $i = \sum_{\ell=0}^{t-1} a_{i,\ell} p^\ell$. Now, for $i = 1, \dots, p^t$ and $j = 1, \dots, p$ let $A_{ij} = \phi_{i-1}(j-1) \bmod p$. Finally, put $A_{i\,p+1} = (i-1) \bmod p$. Now A is an orthogonal array of type $\mathrm{OA}(p^t, p+1, p, t)$. It should be randomized

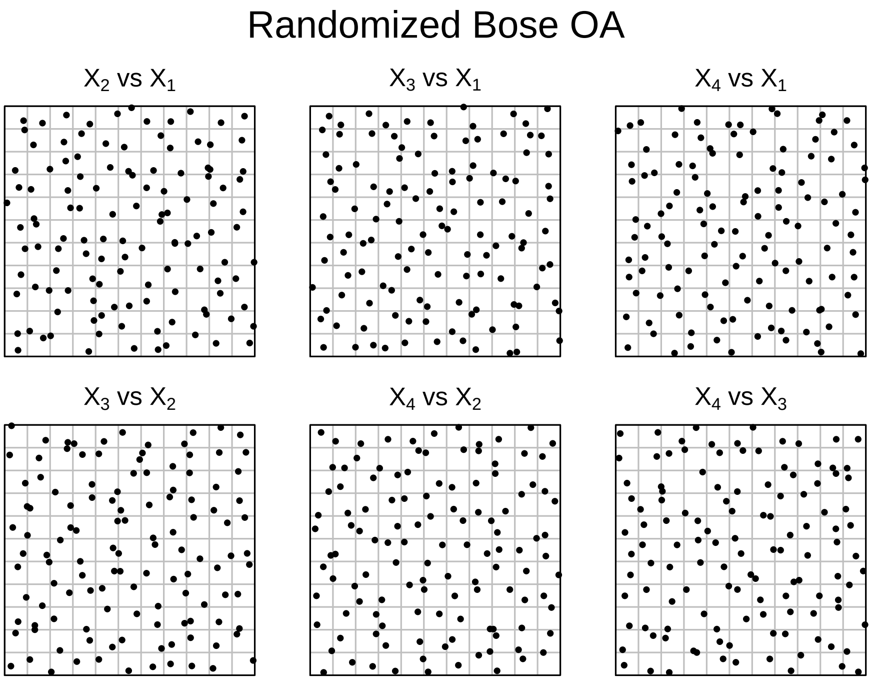

Figure 10.4: This figure shows 121 points of a randomized orthogonal array $\mathrm{OA}(121, 12, 11, 2)$ embedded in the unit cube. All $\binom{12}{2} = 66$ pairwise projections of these points are stratified into a 11×11 grid. The pairs among the first 4 variables are shown.

prior to use in a Monte Carlo application. Bush's proof for this construction is similar to that for the Bose construction above. Once again there is a critical matrix, this time it is the Vandermonde matrix, that we can show is not singular in arithmetic modulo p.

The best strength for randomized orthogonal array sampling is probably $t = 2$ or 3 depending on the problem. Larger values of t start to take on the disadvantages of high dimensional grids.

A disadvantage of randomized orthogonal array sampling, compared to Latin hypercube sampling, is that the univariate margins are only stratified into b intervals not n. Because $b = (n/\lambda)^{1/t}$ is much smaller than n, the additive component of f will be handled less well than with Latin hypercube sampling.

Orthogonal array-based Latin hypercube samples are simultaneously randomized orthogonal arrays and Latin hypercube samples. For the Bose construction, for any given coordinate $j \in \{1, \dots, d\}$ and value $a \in \{0, \dots, b-1\}$, we could find the b values of i for which $A_{ij} = a$ and arrange that the b corresponding U_{ij} values are a stratified sample of $[0, 1)$ (into b intervals of length $1/b$) instead of independent $\mathbf{U}[0, 1)$.

Randomized orthogonal arrays are a stepping stone between Latin hypercube sampling and randomized quasi-Monte Carlo (RQMC) methods presented in Chapter 17. These methods have two ingredients: some raw input points with a balance property, and a randomization which gives those points a $\mathbf{U}[0,1)^d$ distribution, while preserving their balance.

While orthogonal arrays are largely superceded by RQMC they retain some advantages. They have a simpler construction. When we are searching for interesting values of a function $f(\boldsymbol{x})$ the orthogonal array will provide points in all $\binom{d}{t} \times 2^t$ corners of all t-dimensional coordinate projections of $[0,1]^d$ with $n \ll \binom{d}{t} \times 2^t$ points. When, for example, f is singular or negative or otherwise interesting in just a few extreme portions of its domain, randomized orthogonal arrays provide are useful for detecting those locations. The centered version (10.10) is advantageous for graphical exploration of high dimensional functions.

10.5 Adaptive importance sampling

In ordinary importance sampling (Chapter 9) we sample $\boldsymbol{X}_i \overset{\text{iid}}{\sim} q$ and estimate $\mu = \mathbb{E}_p(f(\boldsymbol{X}))$ by

$$\hat{\mu}_q = \frac{1}{n}\sum_{i=1}^n w(\boldsymbol{X}_i)f(\boldsymbol{X}_i), \quad \text{for} \quad w(\boldsymbol{X}_i) = \frac{p(\boldsymbol{X}_i)}{q(\boldsymbol{X}_i)}.$$

Self-normalized importance sampling uses the estimate

$$\widetilde{\mu}_q = \sum_{i=1}^n \widetilde{w}_i f(\boldsymbol{X}_i), \quad \text{where} \quad \widetilde{w}_i = \frac{w(\boldsymbol{X}_i)}{\sum_{j=1}^n w(\boldsymbol{X}_j)}.$$

When either p or q or both are unnormalized we can only compute an unnormalized version $w_u(\boldsymbol{X})$ of w but the unknown normalization factor cancels when we compute $\widetilde{w}_i$.

As we saw in Chapter 9, the choice of q can be a delicate matter. A poor choice could yield an infinite variance for the estimate of μ.

A common practice for choosing q for ordinary importance sampling is to reflect on the nature of p and f and seek a density q that is nearly proportional to pf (when $f \geqslant 0$ everywhere). This q should also have tails that are at least as heavy as those of p. This approach usually takes some informal iteration and may be time consuming and, in difficult settings, we might never find a very good q.

After having sampled $\boldsymbol{X}_i \overset{\text{iid}}{\sim} q$ we learn something about how effective q was. From the resulting data we can estimate the variance of $\hat{\mu}_q$ and, via what-if simulations (§9.14), we can even estimate the variance we would have gotten from some different importance distribution.

The possibility of finding a better importance distribution from q allows us to implement adaptive importance sampling (AIS) schemes. In AIS we alternate

Algorithm 10.1 Generic adaptive importance sampling

```
given nominal density $p$, set $\mathcal{Q}$ of importance densities, integrands $f_1, \dots, f_M$
$k \leftarrow 0$
done $\leftarrow$ FALSE
while not done do
  $k \leftarrow k + 1$
  choose $q^{(k)} \in \mathcal{Q}$ and $n_k \geqslant 1$
  $\boldsymbol{X}_i^{(k)} \overset{\text{iid}}{\sim} q^{(k)}$, $i = 1, \dots, n_k$
  choose done $\in \{\text{TRUE}, \text{FALSE}\}$
choose weights $w(\boldsymbol{X}_1^{(1)}), \dots, w(\boldsymbol{X}_{n_k}^{(k)})$
```

Generic algorithm for AIS. To instantiate it requires choosing importance distributions $q^{(k)}$, sample sizes n_k, a stopping rule and weights on all of the generated samples.

between sampling $\boldsymbol{X}_i^{(k)} \overset{\text{iid}}{\sim} q^{(k)}$ and using the sampled values to choose an improved importance distribution $q^{(k+1)}$. The promise of AIS is that it automates the difficult and subjective task of selecting q, saving considerable labor.

Algorithm 10.1 gives an outline of an AIS algorithm. Turning the AIS idea into an algorithm requires us to make a number of choices. We need to select a family $\mathcal{Q}$ from which to pick the iterates $q^{(k)}$ and an update scheme. Given those, we have to allocate our effort. We might want a large number of small steps, perhaps as small as $n_k = 1$ observation at step k, or a smaller number of steps with larger n_k. The choice depends on how expensive it is to update from $q^{(k)}$ to $q^{(k+1)}$. We might want to spend most of our effort on the last stage with presumably the best $q^{(k)}$. Or we might want to allocate a lot of effort to sampling from $q^{(1)}$, in case the algorithm cannot recover from a bad start. In addition to these choices, we need to pick a stopping rule. Once we have stopped we have to decide how to weight the data from all the iterations. One approach is to only use the very last sample, and another is to weight data from all iterations equally. We will also want confidence intervals on the final estimates.

Some of these choices are difficult to make, not easily automated, and may require some user intervention to tune. Our choices may depend on whether we are seeking a single integral $\int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, a small set of such integrals, or a weighted distribution approximating p for general use. While AIS provides computer-assisted importance sampling, it is not always completely automatic.

In very hard problems even Algorithm 10.1 must be treated as a building block and not as the full algorithm. For example, if $f(\boldsymbol{x}) = \mathbb{1}_{A(\boldsymbol{x})}$ for a very rare event A, our first sample might not contain any instances of A. Then an adaptive algorithm that tried to find q more nearly proportional to $|f|p$ would have no information to go on. In that case, we may define a sequence of easier problems defined by somewhat less rare events than A. If we use AIS along that sequence then Algorithm 10.1 becomes the inner loop of our algorithm. If we use some other algorithm to get started, then AIS becomes the second stage in a two-stage algorithm.

In this section we survey some AIS schemes where $\mathcal{Q}$ is a finite dimensional family of distributions. We focus mainly on how they select $\mathcal{Q}$ and how they update $q^{(k+1)} \in \mathcal{Q}$. Section §10.6 looks at nonparametric families $\mathcal{Q}$.

The adaptive methods are not guaranteed to succeed. An AIS strategy may find a density q that is a good approximation to one mode of fp or, in the self-normalized case, one mode of p.

AIS for self-normalized importance sampling

In self-normalized importance sampling we are not usually trying to estimate probabilities of rare events. We are more usually looking for a distribution q that we can sample from and which provides a close approximation to p. Usually we work with p_u, an unnormalized version of p.

Letting $\mathcal{Q} = \{q(\cdot;\theta) \mid \theta \in \Theta \subseteq \mathbb{R}^r\}$ our problem becomes one of choosing θ. In this context we almost never have $p \in \mathcal{Q}$. We use $\mathbb{E}_\theta(\cdot)$ as a shorthand for $\mathbb{E}_{q(\cdot;\theta)}$.

A very basic way to make q match p is for it to have the same expectation. We might get a better fit by matching both the mean and variance-covariance matrix of $q(\cdot;\theta)$ to those of p. More generally, let $h(\boldsymbol{x}) \in \mathbb{R}^J$ be a vector of functions for which we want $\mathbb{E}_\theta(h(\boldsymbol{x})) \doteq \mathbb{E}_p(h(\boldsymbol{X}))$. For example, to match the mean and variance-covariance matrix we could use

$$h(\boldsymbol{x}) = (x_1, \dots, x_d,\, x_1^2, \dots, x_d^2,\, x_1x_2, \dots, x_{d-1}x_d).$$

When using this method we should choose a family $\mathcal{Q}$ for which $\mathbb{E}_\theta(h(\boldsymbol{X}))$ has a convenient closed form. Often $\theta = \mathbb{E}_\theta(h(\boldsymbol{X}))$, as for example with $q(X;\theta) = \mathcal{N}(\theta, I)$ and $h(\boldsymbol{X}) = \boldsymbol{X}$.

We can use sample values $\boldsymbol{X}_i^{(k)}, \dots, \boldsymbol{X}_{n_k}^{(k)}$ to form the self-normalized estimate of $\mathbb{E}_p(h(\boldsymbol{X}))$

$$\widetilde{\mathbb{E}}^{(k)}(h) = \frac{\sum_{i=1}^{n_k} w_{u,i}^{(k)} h(\boldsymbol{X}_i^{(k)})}{\sum_{i=1}^{n_k} w_{u,i}^{(k)}}$$

for weights $w_{u,i}^{(k)} = p_u(\boldsymbol{X}_i^{(k)})/q(\boldsymbol{X}_i^{(k)};\theta^{(k)})$. If we want to combine all samples from the first k steps then we can use

$$\widetilde{\mathbb{E}}^{(1:k)}(h) = \frac{\sum_{s=1}^{k}\sum_{i=1}^{n_s} w_{u,i}^{(s)} h(\boldsymbol{X}_i^{(s)})}{\sum_{s=1}^{k}\sum_{i=1}^{n_s} w_{u,i}^{(s)}}$$

instead.

The AIS update can be either

$$\theta^{(k+1)} \leftarrow \arg\min_{\theta\in\mathcal{Q}} \|\widetilde{\mathbb{E}}^{(k)}(h) - \mathbb{E}_\theta(h(\boldsymbol{X}))\|, \quad \text{or} \tag{10.12}$$

$$\theta^{(k+1)} \leftarrow \arg\min_{\theta\in\mathcal{Q}} \|\widetilde{\mathbb{E}}^{(1:k)}(h) - \mathbb{E}_\theta(h(\boldsymbol{X}))\|. \tag{10.13}$$

We call (10.12) the ***recent update*** and (10.13) the ***cumulative update***.

If we are using a family with $\mathbb{E}_\theta(h(\boldsymbol{X})) = \theta$ then the recent and cumulative updates simplify to

$$\theta^{(k)} \leftarrow \widetilde{\mathbb{E}}^{(k)}(h), \quad \text{and} \quad \theta^{(k)} \leftarrow \widetilde{\mathbb{E}}^{(1:k)}(h)$$

respectively. This spares us the task of defining and minimizing an appropriate distance $\|\cdot\|$ to use in (10.12) and (10.13).

The recent update (10.12) makes sense if we think that $\theta^{(k)}$ is much better than all the previous parameter values and if n_k is reasonably large. The cumulative update (10.13) has the advantage that it can be used with very small n_k, even as small as $n_k = 1$. The numerator and denominator of $\widetilde{\mathbb{E}}^{(1:k)}(h)$ can be easily updated when new data come in.

The recent update is more stable when $q(\cdot, \theta)$ are not normalized. For unnormalized q the ratios of normalizing constants in $q(\cdot, \theta)$ will affect the relative weighting of different stages of the algorithm. To see the issue, suppose that q is an unormalized $\mathcal{N}(\theta_1, \theta_2^2)$ density. Leaving the $\sqrt{2\pi}$ out of the denominator would give us $q(x, \theta) = \exp(-\frac{1}{2}(x - \theta_1)^2/\theta_2^2)/\theta_2$. We might have used $\exp(-\frac{1}{2}(x - \theta_1)^2/\theta_2^2)$ instead. The recent update is the same either way. But if the value of θ_2 changes sharply as the iterations proceed then the cumulative update may use unpredictable relative weights on the output from different stages.

The cross-entropy method

Now we consider a problem where the the goal is to estimate $\mu = \int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ with $f(\boldsymbol{x}) \geqslant 0$ and μ strictly greater than 0. In this case there exists a zero-variance importance distribution $q \propto fp$, although it is usually not in our family $\mathcal{Q}$. We assume that p and $q(\cdot;\theta)$ are both normalized. Then we may use the estimate

$$\hat{\mu}_\theta = \frac{1}{n}\sum_{i=1}^n w_\theta(\boldsymbol{X}_i)f(\boldsymbol{X}_i)$$

where $w_\theta(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x};\theta)$. The optimal density from within our family is the one that minimizes

$$v(\theta) \equiv \int \frac{f(\boldsymbol{x})^2 p(\boldsymbol{x})^2}{q(\boldsymbol{x};\theta)}\,\mathrm{d}\boldsymbol{x}.$$

Given a sample $\boldsymbol{X}_i^{(k)}$ from $q(\cdot;\theta^{(k)})$, we can estimate $v(\theta)$ by

$$\hat{v}^{(k)}(\theta) = \frac{1}{n}\sum_{i=1}^n \frac{f(\boldsymbol{X}_i^{(k)})^2 p(\boldsymbol{X}_i^{(k)})^2}{q(\boldsymbol{X}_i^{(k)};\theta)q(\boldsymbol{X}_i^{(k)};\theta^{(k)})}.$$

In principal, we might minimize $\hat{v}^{(k)}$ over θ to find $\theta^{(k+1)}$. In practice, this may be difficult to do. The form of q may not be convenient for optimization. Furthermore, as we saw in §9.14, the sample from $\theta^{(k)}$ might only provide good estimates for θ near $\theta^{(k)}$.

The ***cross-entropy*** method makes two simplifications to enable a more convenient update. First, it replaces variance by the Kullback-Leibler distance presented below. Second, it works with a family $\mathcal{Q}$ defined by exponential tilting. We will take

$$q(\boldsymbol{x};\theta) = \exp(\theta^{\mathsf{T}}\boldsymbol{x} - A(\boldsymbol{x}) - C(\theta)). \tag{10.14}$$

This makes $\mathcal{Q}$ a natural exponential family. In a more general exponential family, the term $\theta^{\mathsf{T}}\boldsymbol{x}$ is generalized to $\eta(\theta)^{\mathsf{T}}T(\boldsymbol{x})$ for functions $\eta(\cdot)$ and $T(\cdot)$. The combination of Kullback-Leibler distance and a natural exponential family will yield a closed-form update for θ.

The ***Kullback-Leibler distance*** from density g to h is

$$\mathcal{D}(g\|h) = \mathbb{E}_g\Bigl(\log\Bigl(\frac{g(\boldsymbol{X})}{h(\boldsymbol{X})}\Bigr)\Bigr).$$

It is not a proper distance in that it lacks symmetry: $\mathcal{D}(g\|h) \neq \mathcal{D}(h\|g)$ in general. But it does satisfy $\mathcal{D}(g\|g) = 0$. Also, by convexity of $-\log(\cdot)$ and Jensen's inequality

$$\mathcal{D}(g\|h) = \mathbb{E}_g(-\log(h/g)) \geqslant -\log(\mathbb{E}_g(h/g)) = 0.$$

We will interpret $\mathcal{D}(g\|h)$ as a measure of how far h is from g. Although we describe K-L and cross-entropy for probability density functions they also apply to probability mass functions. The cross-entropy method takes its name from the function $-\mathbb{E}_g(\log(h))$, which is called the cross-entropy.

The optimal q is $q^* = fp/\mu$. Now

$$\mathcal{D}(q^*\,\|\,q(\cdot;\theta)) = \mathbb{E}_{q^*}\bigl(\log(q^*(\boldsymbol{X})) - \log(q(\boldsymbol{X};\theta))\bigr)$$

and so to minimize $\mathcal{D}(q^*\|q(\cdot;\theta))$ we maximize $\mathbb{E}_{q^*}(\log(q(\boldsymbol{X};\theta)))$. Let q be a distribution such that $q(\boldsymbol{x}) > 0$ whenever $q^*(\boldsymbol{x}) > 0$. Then our optimal θ maximizes

$$\mathbb{E}_q\Bigl(\frac{q^*(\boldsymbol{X})}{q(\boldsymbol{X})}\log(q(\boldsymbol{X};\theta))\Bigr) = \mu\mathbb{E}_q\Bigl(\frac{p(\boldsymbol{X})f(\boldsymbol{X})}{q(\boldsymbol{X})}\log(q(\boldsymbol{X};\theta))\Bigr).$$

The constant $\mu > 0$ does not affect the optimizer.

Given data $\boldsymbol{X}_i^{(k)} \sim q(\cdot;\theta^{(k)})$ the cross-entropy method chooses $\theta^{(k+1)}$ to maximize

$$\begin{aligned} G^{(k)}(\theta) &= \frac{1}{n_k}\sum_{i=1}^{n_k}\frac{p(\boldsymbol{X}_i^{(k)})f(\boldsymbol{X}_i^{(k)})}{q(\boldsymbol{X}_i^{(k)};\theta^{(k)})}\log(q(\boldsymbol{X}_i^{(k)};\theta)) \\ &\equiv \frac{1}{n_k}\sum_{i=1}^{k} H_i\log(q(\boldsymbol{X}_i^{(k)};\theta)), \end{aligned} \tag{10.15}$$

over θ, where we use $H_i^{(k)} \geqslant 0$ to soak up computable quantities that do not depend on θ. Equation (10.15) contains three distributions: the nominal density

p, the density $q(\cdot,\theta^{(k)})$ from which the most recent data were sampled, and the candidate density $q(\cdot;\theta)$ for the next iteration.

When $\mathcal{Q}$ is a natural exponential family then

$$\log(q(\boldsymbol{X}_i^{(k)};\theta)) = \theta^\mathsf{T}\boldsymbol{X}_i^{(k)} - A(\boldsymbol{X}_i^{(k)}) - C(\theta)$$

where $C(\cdot)$ is known to be a convex function of θ. Then (10.15) becomes

$$G^{(k)}(\theta) = \frac{1}{n_k}\sum_{i=1}^{k} H_i^{(k)}\bigl(\theta^\mathsf{T}\boldsymbol{X}_i^{(k)} - A(\boldsymbol{X}_i^{(k)}) - C(\theta)\bigr).$$

It is possible that all the $H_i^{(k)} = 0$ and then maximizing (10.15) cannot work. For now we suppose that $\sum_i H_i^{(k)} > 0$.

Because $C(\theta)$ is convex, $G^{(k)}$ is a concave function of θ. When $C(\cdot)$ is differentiable we may then find $\theta^{(k+1)}$ by setting $\partial G^{(k)}(\theta)/\partial\theta = 0$. Then $\theta^{(k+1)}$ solves

$$\frac{\sum_i H_i^{(k)}\boldsymbol{X}_i^{(k)\mathsf{T}}}{\sum_i H_i^{(k)}} = \frac{\partial}{\partial\theta}C(\theta^{(k+1)}). \tag{10.16}$$

There is a transpose on $\boldsymbol{X}_i^{(k)}$ to reflect the convention that $\boldsymbol{X}$ are column vectors and the gradient of the scalar function C is a row vector. Equation (10.16) sets a weighted average of the $\boldsymbol{X}_i^{(k)}$ equal to the gradient of $C(\cdot)$ at the new iterate $\theta^{(k+1)}$.

To see how this works, suppose that $q(\boldsymbol{x};\theta)$ is the $\mathcal{N}(\theta,\Sigma)$ distribution for some fixed nonsingular matrix Σ. Then after some rearrangement

$$q(\boldsymbol{x};\theta) = \exp(\theta^\mathsf{T}\boldsymbol{x} - \tfrac{1}{2}\boldsymbol{x}^\mathsf{T}\Sigma^{-1}\boldsymbol{x} - \tfrac{1}{2}\theta^\mathsf{T}\Sigma^{-1}\theta - \tfrac{1}{2}\log|\Sigma| - (d/2)\log(2\pi)).$$

Matching with (10.14) we find that $C(\theta) = \frac{1}{2}\theta^\mathsf{T}\Sigma^{-1}\theta$ from which $\partial/\partial\theta C(\theta) = \theta^\mathsf{T}\Sigma^{-1}$ and so the update is

$$\theta^{(k+1)} \leftarrow \Sigma \times \frac{\sum_i H_i^{(k)}\boldsymbol{X}_i^{(k)}}{\sum_i H_i^{(k)}}. \tag{10.17}$$

That is, one uses (10.17) to choose the next density q in Algorithm 10.1.

For another example, suppose that $\boldsymbol{X} \in \{0,1\}^d$ with independent but not identically distributed components. We will use the logit function $\mathrm{logit}(\eta) = \log(\eta/(1-\eta))$. Then the probability mass function of $\boldsymbol{x}$ is

$$\prod_{j=1}^{d}\eta_j^{x_j}(1-\eta_j)^{1-x_j} = \exp\biggl(\sum_{j=1}^{d} x_j\mathrm{logit}(\eta_j) + \log(1-\eta_j)\biggr).$$

If we adopt the parameter $\theta \in \mathbb{R}^d$ with $\theta_j = \mathrm{logit}(\eta_j)$ then we can put the distribution into the natural exponential form

$$q(\boldsymbol{x};\theta) = \exp\biggl(\boldsymbol{x}^\mathsf{T}\theta - \sum_{j=1}^{d}\log(1+e^{\theta_j})\biggr).$$

Algorithm 10.2 Cross-entropy for $\mathbb{P}(f(\boldsymbol{X}) \geqslant \tau)$

```
given family $\mathcal{Q} = \{q(\cdot;\theta), \theta \in \Theta\}$, $p(\cdot)$, $f(\cdot)$, $\tau$, $n$, $K$, $\theta^{(1)}$, $\varepsilon$
for $k = 1, \dots, K$ do
    $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \overset{\text{iid}}{\sim} q(\cdot;\theta^{(k)})$
    $\tau^{(k)} \leftarrow (1-\varepsilon)$-quantile of $f(\boldsymbol{X}_1), \dots, f(\boldsymbol{X}_n)$
    $\tau^{(k)} \leftarrow \min(\tau, \tau^{(k)})$
    $w_i \leftarrow p(\boldsymbol{X}_i)/q(\boldsymbol{X}_i;\theta^{(k)}), \quad i = 1, \dots, n$
    $H_i \leftarrow w_i \mathbb{1}_{f(\boldsymbol{X}_i) \geqslant \tau^{(k)}}, \quad i = 1, \dots, n$
    $\theta^{(k+1)} \leftarrow \arg\max_\theta \sum_{i=1}^n H_i \log(q(\boldsymbol{X}_i;\theta))$
$\hat\mu^{(K)} \leftarrow \frac{1}{n}\sum_{i=1}^n w_i \mathbb{1}_{f(\boldsymbol{X}_i) \geqslant \tau}$
deliver $\hat\mu^{(K)}$, $\tau^{(K)}$
```

Basic cross-entropy for rare events. Often $\varepsilon = 0.01$ is used. Then $n \geqslant 1000$ is advisable. More sophisticated versions could pool estimates from all stages, attempt to diagnose convergence failures and produce confidence intervals.

In this form we get $C(\theta) = \sum_{j=1}^d \log(1 + e^{\theta_j})$. Then $\partial C/\partial\theta_j = e^{\theta_j}/(1 + e^{\theta_j})$, the inverse of the logit function. The cross-entropy update is

$$\theta^{(k+1)} \leftarrow \text{logit}\left(\frac{\sum_i H_i^{(k)} \boldsymbol{X}_i^{(k)}}{\sum_i H_i^{(k)}}\right),$$

interpreted componentwise.

The cross-entropy update is not well defined if all of the $H_i^{(k)}$ are zero. This may happen if $f(\boldsymbol{X}_i^{(k)}) = 0$ for $i = 1, \dots, n_k$. If $f(\boldsymbol{x}) = \mathbb{1}_{S(\boldsymbol{x}) \geqslant \tau}$ for a very high threshold τ, then we might expect our first sample to contain no points with $f \neq 0$. The cross-entropy method can recover from this problem by using a smaller value of τ. For instance, if all $S(\boldsymbol{X}_i^{(k)}) < \tau$ we may replace τ by a lower value $\tau^{(k)}$ such as the 99'th percentile of $\boldsymbol{X}_1^{(k)}, \dots, \boldsymbol{X}_{n_k}^{(k)}$. The next iteration should generate larger values of $S(\boldsymbol{X})$ with a larger 99'th percentile. We may use the larger of the original τ or the 99'th percentile from step k to define θ for step $k+1$. Ideally, the value $\tau^{(k)}$ reaches τ early on and remains there.

The cross-entropy method is one of the easiest adaptive importance sampling methods to use. It tilts the distribution of $\boldsymbol{X}$ in a direction θ and chooses θ with little intervention from the user. In Exercise 10.7 there is only one direction that brings extreme values for $f(\boldsymbol{X})$ and the cross-entropy method finds it very efficiently and gives a good estimate for a very small event probability.

The cross-entropy method can go astray. If we tilt towards one source of extreme values, we may well tilt away from another and miss that source's contribution to μ. Exercise 10.8 provides a cautionary tale.

Exponential convergence

In some settings $f(\boldsymbol{x}) \geqslant 0$ and the optimal density $q^* = pf/\mu$ belongs to our set $\mathcal{Q}$. There is then a zero variance sampling density among our choices for some

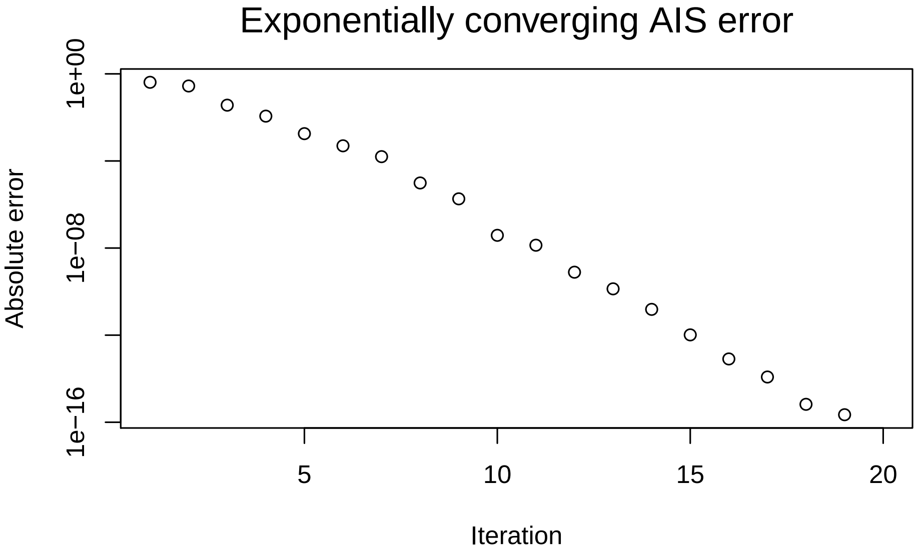

Figure 10.5: Toy example where absolute error converges exponentially to zero in AIS.

value θ^*. Sometimes we can get an estimate $\hat\theta$ from n observations with RMSE $(\mathbb{E}((\hat\theta-\theta^*)^2))^{1/2} = O(1/\sqrt{n})$. Sampling from a better $\hat\theta$ may then lead to an even better estimate of $\hat\theta$. The result may be an exponential convergence where $\exp(\kappa k)(\hat\mu^{(k)}-\mu)$ converges to 0 as k increases for some $\kappa > 0$.

We can illustrate the phenomenon in a toy example where we would not use AIS. This toy example will even use the known relationship between μ and θ to do the updates. Let $p(x) = \mathcal{N}(0,1)$, $f(x) = \exp(Ax)$ for $A > 0$ and $q(x;\theta) = \mathcal{N}(\theta,1)$. We then find that $\mu = \exp(A^2/2)$ and proportionality $fp = q(\cdot;\theta)\mu$ holds for $\theta = \theta^* = A$. Because $A > 0$, we have $A = \sqrt{2\log(\mu)}$.

Since the optimal sampler has $\theta = A$ and $\mathbb{E}_p(f(X)) = \exp(A^2/2)$ we may consider an AIS scheme starting with $\theta^{(1)} = 0$, and then for $k \geqslant 1$ updating by

$$\begin{aligned}
X_i^{(k)} &\overset{\mathrm{iid}}{\sim} q(\cdot;\theta^{(k)}), \quad i = 1,\dots,n,\\
\hat\mu^{(k)} &\leftarrow \frac{1}{n}\sum_{i=1}^n \frac{f(X_i^{(k)})p(X_i^{(k)})}{q(X_i^{(k)};\theta^{(k)})}, \quad \text{and}\\
\hat\theta^{(k+1)} &\leftarrow \sqrt{2\log(\hat\mu^{(k)})}.
\end{aligned} \tag{10.18}$$

Figure 10.5 illustrates this update for steps $k = 1,\dots,20$ with $n = 20$ and $A = 1$. The log absolute error versus iteration is nearly linear and by $k = 20$ the error has reached 0 in double precision.

The AIS update illustrated in Figure 10.5 is delicate in one way. When run with $n = 3$ it was observed to give an estimate $\hat\mu^{(1)} < 1$. In that case $\theta^{(2)}$ is not well defined. We know that in this case $\mu > 1$. A better update would be

$$\hat{\theta}^{(k+1)} \leftarrow \sqrt{2\log(\max(1, \hat{\mu}^{(k)}))}. \tag{10.19}$$

This is an example of the special attention we may have to pay to the starting conditions for AIS.

The simulation in Figure 10.5 used $K = 20$ steps of $n = 20$ observations. A smaller sample size n might allow the simulation to adapt more quickly. On the other hand it might be less stable. Exercise 10.10 looks at the algorithm in (10.18) for $K = 400$ steps of size $n = 1$ using (10.19).

Exponential or near-exponential convergence is seen is in some Monte Carlo calculations used in particle transport. Example problems include simulations to measure the effectiveness of sensors or shields in nuclear reactors or in medical imaging. The goal is to compute a quantity like the average energy of particles reaching a target area. When the particles don't interfere with each other or change the material that they pass through, then it is enough to find the average behavior of one random particle and multiply that by the number of particles.

The particle moves through a material such as a lead or concrete shield. It will fly through the material for some distance until it either leaves the region of interest, enters a region made of a different material, or collides with an atom. In a collision it could be absorbed. Or it could lose some energy and resume its journey in a new direction. The important properties of the particle are its location in three dimensions, its velocity which is also three dimensional, and its energy level, for a total state space of seven dimensions.

A direct simulation from the particle transport distributions is not effective here because the probability of a nonzero score is extremely small. Therefore, some kind of importance sampling is in order. Intuitively, we could sample particles with increased energy, or give them longer trajectories or bias them in the direction of the shielded region. The optimal approach is likely some combination of these and maybe other strategies.

We can simplify the problem by supposing that the particle may be in one of d states, $\{1, 2, \dots, d\}$. Because the state space is seven dimensional we might need a large d to cover it adequately. After presenting the basic idea, we will look at ways to mitigate the large state space. The description here follows Kollman et al. (1999) and other references given in the end notes.

Once the particle is absorbed or leaves the region of interest, we don't have to follow it any more. Instead, we suppose that it has entered a terminal state Δ. Once the particle reaches state Δ, it remains there forever.

The path the particle takes is described by a time homogeneous Markov chain. Some background on Markov chains is in Chapter 11 on Markov chain Monte Carlo. Letting X_n record the state that the particle is in at step n, the homogeneous Markov property implies that $\mathbb{P}(X_{n+1} = x \mid X_n, X_{n-1}, \dots, X_0) = \mathbb{P}(X_{n+1} = x \mid X_n)$. For $i, j \in \{1, \dots, d\}$ we let $P_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i)$. The probability of a transition from state i to Δ is $P_{i\Delta} \equiv 1 - \sum_{j=1}^{d} P_{ij}$. Though we write $P_{i\Delta}$, we will use P itself to refer to just the $d \times d$ matrix of non-terminal transitions.

In these problems, the matrix $P \in [0,1]^{d\times d}$ satisfies $\lim_{n\to\infty} P^n = 0$ and it then follows that $\lim_{n\to\infty} \mathbb{P}(X_n = \Delta \mid X_0 = i) = 1$ for all $i \in \{1,\dots,d\}$. The particle will not bounce forever but must eventually be absorbed or leave the region of interest. Let

$$\tau = \min\{n \geqslant 1 \mid X_n = \Delta\}$$

be the hitting time of Δ. We have $\mathbb{P}(\tau < \infty \mid X_0 = i) = 1$ for all i.

To capture the measurement of interest we define a score function s_{ij} such that a transition from state i to state j adds the score $s_{ij} \geqslant 0$ to that particle's total. For example, $s_{i\Delta}$ may be the amount of energy hitting a target region when a particle is absorbed from state i. Similarly, s_{ij} may be the energy left in the target region when the particle loses energy due to a collision. Particles in the terminal state do not increase their score: $s_{\Delta\Delta} = 0$. When it is typographically more convenient, we write $s(i \to j)$ instead of s_{ij}. The total score from a particle's trajectory is

$$Y = \sum_{n=1}^{\infty} s(X_{n-1} \to X_n) = \sum_{n=1}^{\tau} s(X_{n-1} \to X_n),$$

and we want to find $\mu_i = \mathbb{E}_p(Y \mid X_0 = i)$ for all or some of $i = 1,\dots,d$.

We will importance sample replacing the matrix P by another one, Q. We require that $Q_{ij} > 0$ whenever $P_{ij} > 0$. We assume this even if $s_{ij} = 0$, in case a later score along the path leaving j is positive. We also require $Q_{i\Delta} > 0$ whenever $P_{i\Delta} > 0$. The likelihood ratio appropriate to the n'th transition is $w_n(X) = \prod_{j=1}^n P_{X_{j-1}X_j}/Q_{X_{j-1}X_j}$. If we sample $\boldsymbol{X}_i$ from Q, then

$$\mathbb{E}_q\biggl(\sum_{n=1}^{\tau} s(X_{n-1} \to X_n) w_n(X) \mid X_0 = i\biggr) = \mu_i$$

providing a basis for importance sampled estimation of μ_i.

There is a zero variance importance sampling distribution for the vector $\boldsymbol{\mu} = (\mu_1,\dots,\mu_d)$. The optimal transition from i to j takes place with probability proportional to P_{ij} times the expected score of the ij transition and all future transitions. The immediate score is s_{ij} and having landed in state j, the expected future score is μ_j. For $i,j \in \{1,\dots,d\}$, the zero variance importance sampler has transitions

$$Q_{ij}(\boldsymbol{\mu}) = \frac{P_{ij}(s_{ij} + \mu_j)}{P_{i\Delta}s_{i\Delta} + \sum_{\ell=1}^d P_{i\ell}(s_{i\ell} + \mu_\ell)}, \quad \text{and}$$
$$Q_{i\Delta}(\boldsymbol{\mu}) = \frac{P_{i\Delta}s_{i\Delta}}{P_{i\Delta}s_{i\Delta} + \sum_{\ell=1}^d P_{i\ell}(s_{i\ell} + \mu_\ell)}.$$

As usual, the optimal sampler depends on the unknown $\boldsymbol{\mu}$. Adaptive importance sampling here alternates between sampling M trajectories from $Q(\hat{\boldsymbol{\mu}}^{(k)})$ starting at each state $X_0 \in \{1,\dots,d\}$ and computing $\hat{\boldsymbol{\mu}}^{(k+1)}$ from those trajectories.

There is a hazard here with an interesting resolution. Many of the $s_{ij} = 0$. It is possible to get an estimate $\hat\mu_j^{(k)} = 0$ even when $\mu_j > 0$. That would have the very undesirable consequence of making $Q_{ij}(\hat{\boldsymbol{\mu}}^{(k)}) = 0$ and the algorithm might never recover from wrongly making $i \to j$ transitions impossible. We have to bound all of the $\hat\mu_i^{(k)}$ away from zero. The solution is to artificially add some $\delta > 0$ to every $s_{i\Delta}$. Every walk terminates with a step to Δ and so then all $\mu_i \geqslant \delta$ and all $\hat\mu_i > 0$. When the algorithm finishes we can subtract δ from each estimated μ_i.

The algorithm starts with every $\hat\mu_i^{(0)} > \delta > 0$. To update from $\boldsymbol{\mu}^{(k)}$ to $\boldsymbol{\mu}^{(k+1)}$ we run M simulated paths starting at $X_0 = i$ for every state $i = 1, \dots, d$ using $Q(\boldsymbol{\mu}^{(k)})$ as transition matrix, and let

$$\bar Y_i^{(k)} = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{\tau_m^{(k)}} s\bigl(X_{mn}^{(k)} \to X_{mn+1}^{(k)}\bigr) \times w_n(X_m^{(k)}), \quad \text{for } X_{m0}^{(k)} = i.$$

Then we take

$$\hat\mu_i^{(k+1)} = \max(\bar Y_i^{(k)}, \delta), \quad i = 1, \dots, d. \tag{10.20}$$

This estimate is biased but never less accurate than the unbiased estimate $\bar Y_i^{(k)}$ because we have arranged for $\mu_i \geqslant \delta$.

Theorem 10.3. *Let the matrix P satisfy $\lim_{n\to\infty} P^n = 0$. Assume that there is a $\delta > 0$ with $s_{i\Delta} \geqslant \delta$ for $i = 1, \dots, d$. Let $\hat\mu^{(k+1)}$ be given by* (10.20) *starting with all $\mu_i^{(0)} > \delta > 0$. Then there exist constants $\kappa > 0$ and M_0 such that*

$$\lim_{k\to\infty} \mathbb{P}(e^{\kappa k}\|\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}\| \to 0) = 1$$

if $M \geqslant M_0$.

Proof. This is Theorem 3.2 of Kollman (1993). □

Theorem 10.3 verifies that exponential convergence is possible for AIS. In practice, we have to contend with the large state space. Sometimes it is possible to reduce the physical space to one dimension representing how far through a planar shield the particle has traveled. Then the state space has three dimensions, depth, velocity in the retained direction and energy. Another approach is to work in a continuous state space but with the function $\mu(\cdot) = \mathbb{E}(Y \mid X_0 = \cdot)$ on that space approximated within a finite dimensional parametric family. We then only need to use a number of starting points X_0 as large as or somewhat larger than the dimension of this family. In such cases there are theorems showing exponential convergence when the true $\mu(\cdot)$ really has the assumed parametric form. Empirical results often show initial exponential convergence followed by slower convergence, when lack of fit to the functional form comes to dominate the errors. The estimates remain unbiased, but fail to be from a zero variance estimator. See page 53 of the end notes for references.

10.6 Nonparametric AIS

Nonparametric versions of AIS use larger and more flexible families $\mathcal{Q}$ of importance sampling candidate densities. They usually have parameters, but perhaps a large or indefinite number of them. A common strategy is to let q be a piece-wise constant density, that is, a mixture of uniform distributions. Adaptive importance sampling with a mixture of uniform distributions amounts to adaptive stratification. It is convenient to work in a setting where $\boldsymbol{X} \sim \mathbf{U}(0,1)^d$ under the nominal distribution p. Many problems involving non-uniform variables can be accomodated through a transformation $\phi(\boldsymbol{X})$.

It is usual for q to be constant in hyper-rectangular subregions of $(0,1)^d$. The algorithms alternate between strategic partitioning of the input space and sampling from the resulting mixture distribution.

Here we will study the main ideas behind a few nonparametric AIS algorithms. Those ideas are of interest because they can be built in to new algorithms or guide our choice of algorithm for a specific problem. Implementations of these algorithms involve a number of technical decisions and even some ad hoc choices to safeguard against problems that their authors anticipated. We will not cover all of those issues. They are needed if we want to duplicate the algorithms precisely, but in that case it is better to consult the original research or obtain the original authors' code.

Vegas

The Vegas algorithm (Lepage, 1978) is an adaptive importance sampling method used in physics. Sometimes the term ***Las Vegas algorithm*** is to describe an algorithm that uses randomness to get an exact result in a possibly random amount of computation. That is a different usage.

Vegas is a complicated algorithm and we will not describe it in full. There is a brief description in Press et al. (2007, Chapter 7) which includes the source code with some added comments. There are more details in the technical report Lepage (1980). This account is based in part on Zhou's (1998) very thorough description.

Vegas works with a product density $q(\boldsymbol{x}) = \prod_{j=1}^d q_j(x_j)$. Each factor in the product density is a piece-wise constant function. The distribution q_j is defined by break points $0 = x_{j,0} < x_{j,1} < \cdots < x_{j,N_j} = 1$. Let $\Delta_{j,\ell} = x_{j,\ell} - x_{j,\ell-1}$ for $1 \leqslant \ell \leqslant N_j$. Then

$$q_j(x) = \frac{1}{N_j \Delta_{j,\ell}}, \quad x_{j,\ell-1} \leqslant x < x_{j,\ell}.$$

Instead of splitting $[0,1)$ into N_j equal width pieces and varying their probabilities, the algorithm splits $[0,1)$ into N_j intervals $I_{j,\ell} = [x_{j,\ell-1}, x_{j,\ell})$ of varying width but equal probability content $1/N_j$. That choice is one of Vegas' most important features. It allows Vegas to focus on very narrow modes in f by using very small widths $\Delta_{j,\ell}$ for number of consecutive values of ℓ. It is easy to sample $X_j \sim q_j$ by inversion. For $U \sim \mathbf{U}(0,1)$, we use $\ell \leftarrow \lfloor N_j U \rfloor$ and

$X_j \leftarrow x_{j,\ell} + N_j(U - \ell/N_j)\Delta_{j,\ell}$. Though easy to sample, this parameterization is challenging to update.

The optimal importance sampling distribution is proportional to $|f|$ because p is constant. The optimal importance sampling distribution within the class of product densities has

$$q_j(x_j) \propto \sqrt{\int_{(0,1)^{d-1}} \frac{f(\boldsymbol{x})^2}{\prod_{\ell\neq j} q_\ell(x_\ell)} \prod_{\ell\neq j} \mathrm{d}x_\ell}. \tag{10.21}$$

Given a sample from $q^{(k)}$, Vegas updates its estimates to $q^{(k+1)}$. The first step is to iterate a sample analogue of (10.21) to update the probability content that X_j should place on $I_{j,\ell}$. Let this new value be $p_{j,\ell}$.

Vegas does its updating by moving the endpoints $x_{j,\ell}$ instead of updating the probability content of $I_{j,\ell}$. That is a very complicated process which uses the $p_{j,\ell}$ values to guide the move. It first sets $r_{j,\ell} = p_{j,\ell}\Delta_{j,\ell}$ which is an estimate of the desired probability content for $I_{j,\ell}$. Then it smooths these values getting $\widetilde{r}_{j,\ell}$ a weighted average of $r_{j,\ell}$ and its neighbors $r_{j,\ell-1}$ (if $\ell \geqslant 2$) and $r_{j,\ell+1}$ (if $\ell \leqslant N_j - 1$). The $\widetilde{r}_{j,\ell}$ values are then normalized to sum to 1. This smoothing is designed to reduce variance.

The $\widetilde{r}_{j,\ell}$ values are 'flattened' using the transformation

$$\widetilde{r}_{j,\ell} \leftarrow F(\widetilde{r}_{j,\ell}), \quad \text{where} \quad F(r) \equiv \Bigl(\frac{1-r}{-\log(r)}\Bigr)^{\alpha}$$

(default $\alpha = 1.5$). The flattening transformation F is nearly linear for all but the smallest $\widetilde{r}_{j,\ell}$. It has the effect of raising the smaller $\widetilde{r}_{j,\ell}$ relative to the others, like defensive importance sampling of §9.11 does. Taking limits, $F(0) = 0$ and $F(1) = 1$. The lower limit implies no defensive mixing at $\widetilde{r}_{j,\ell} = 0$ but with $F(10^{-100}) \doteq 0.00029$ we see that even very small $\widetilde{r}_{j,\ell}$ get boosted up.

We may now consider a piecewise constant density function with a value proportional to $\widetilde{r}_{j,\ell}$ in the old interval $I^{(k)}_{j,\ell}$. The updated break point $x^{(k+1)}_{j,\ell}$ is the ℓ/N_j quantile of this mixture of uniforms.

As Vegas runs through K iterations, it retains estimates $\hat\mu^{(k)}$ for $k = 1,\dots,K$ and estimates $\widehat{\mathrm{Var}}(\hat\mu^{(k)})$ of $\mathrm{Var}(\hat\mu^{(k)})$, both based on importance sampling formulas. The final estimate from Vegas is

$$\hat\mu_{\mathrm{Vegas}} = \frac{\sum_{k=1}^K \hat\mu_k/\widehat{\mathrm{Var}}(\hat\mu^{(k)})}{\sum_{k=1}^K 1/\widehat{\mathrm{Var}}(\hat\mu^{(k)})}. \tag{10.22}$$

Vegas also produces a variance estimate

$$\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{Vegas}}) = \Bigl(\sum_{k=1}^K \widehat{\mathrm{Var}}(\hat\mu^{(k)})^{-1}\Bigr)^{-1/2},$$

and a diagnostic statistic

$$\chi^2 = \sum_{k=1}^K \frac{(\hat\mu^{(k)} - \hat\mu_{\mathrm{Vegas}})^2}{\widehat{\mathrm{Var}}\mu^{(k)}}.$$

If χ^2 is significantly larger than $K-1$ it indicates that the K estimates being pooled appear to be estimating different quantities and then Lepage (1978) says the results are not to be trusted. An alternative to the pooling method (10.22) is given on page 34.

The user of Vegas chooses K as well as the number of sample points to be drawn at each iteration. Vegas has a built-in upper bound on the input dimension, $d \leqslant 10$.

Divonne

The Divonne algorithm described in Friedman and Wright (1979, 1981) uses recursive splitting of the cube $[0,1]^d$. The special ingredient in their algorithm is the use of numerical optimization algorithms to guide the partitioning. If f has two continuous derivatives then we can reliably find local minima and maxima of f, often with a modest number of function evaluations. Optimizing within hyperrectangular subsets of $[0,1]^d$ amounts to imposing simple bound constraints on the components x_j and such constraints are built in to many optimizers. Local optima are not necessarily global ones, but for a well-behaved function f and a not too large subregion of $[0,1]^d$ they will coincide. When the local optima are not global, that will affect efficiency but not unbiasedness of the resulting estimated expectation.

Divonne uses multiple random starts each time it seeks a maximum or minimum, to improve the odds of finding the global optimum. It employs a quasi-Newton algorithm using divided differences to estimate gradients, so the user does not have to supply a function to compute the gradient of f.

Divonne maintains a list of hyperrectangular subsets $R_1, \dots, R_L \subset [0,1]^d$ with $\cup_{\ell=1}^L R_\ell = [0,1]^d$ and $\mathbf{vol}(R_j \cap R_k) = 0$ for $j \neq k$. Initially $L = 1$ and $R_1 = [0,1]^d$. We'll refer to these subsets as simply 'rectangular' with the understanding that $d \neq 2$ is included.

Let $R = [\boldsymbol{a}, \boldsymbol{b}] = \prod_{j=1}^d [a_j, b_j]$ be a rectangular subset of $[0,1]^d$. They define a badness, or spread function

$$s(R) = \mathbf{vol}(R) \times \bigl(\max_{\boldsymbol{x}\in R} f(\boldsymbol{x}) - \min_{\boldsymbol{x}\in R} f(\boldsymbol{x})\bigr)$$

for this rectangle. Each step of the algorithm involves replacing the worst rectangle in the list (largest $s(\cdot)$) by two or more sub-rectangles. The selected rectangle is removed from the list and replaced by its sub-rectangles. That operation continues until a convergence criterion based on estimated accuracy is met. Their approach to subdividing R differs from most. Instead of splitting R into two congruent halves they seek the most prominent spike of f within R and find a subrectangle that isolates this spike from the rest of R.

To subdivide R they find $\boldsymbol{x}_{\min}$ and $\boldsymbol{x}_{\max}$, the minimizer and maximizer respectively of f over $\boldsymbol{x} \in R$. They also get a sample based estimate $\hat\mu(R)$ of $\mathbb{E}(f(\boldsymbol{X}) \mid \boldsymbol{X} \in R)$. The partition of R is based on a rectangle $R_M \subset R$ built around $\boldsymbol{x}_M$ which is either the mode at $\boldsymbol{x}_{\max}$ or the minimum at $\boldsymbol{x}_{\min}$, whichever of these has an f value farther from $\hat\mu(R)$.

The new rectangle R_M isolating $\boldsymbol{x}_M$ has the form $\prod_{j=1}^d [x_{Mj} - \delta_j^-, x_{Mj} + \delta_j^+]$ for carefully chosen $\delta_j^\pm$. If x_{Mj} is at or near a_j then $\delta_j^- = x_{Mj} - a_j$. Specifically, if $x_{Mj} - a_j \leqslant \beta(b_j - a_j)$ (default $\beta = 0.05$) then $\delta_j^- = x_{Mj} - a_j$. Similarly, if X_{Mj} is at or near b_j then the new box will have j'th upper limit b_j. When the interval $[x_{Mj}, b_j]$ is splittable a further constraint is that $\delta_j^+ \leqslant \alpha|b_j - x_{Mj}|$ (default $\alpha = 0.5$) and similarly when $[a_j, x_{Mj}]$ is splittable.

When there is some freedom to choose $\delta_j^\pm$ it is used to make $f(x_M + \boldsymbol{e}_j\delta_j^+)$ and/or $f(\boldsymbol{x}_M - \boldsymbol{e}_j\delta_j^-)$ take on a common value $\widetilde{f}$. Here $\boldsymbol{e}_j$ is the unit vector in the component j's direction. The value $\widetilde{f}$ is chosen to make estimates of $s(R_M)$ and $s(R \cap R_M^c)$ match. Having carved out the modal rectangle R_M from within R, the complement $R - R_M$ is now a rectangle minus a rectangular hole. The complement $R - R_M$ is split into a list of rectangles, possibly as many as $2d$ but often fewer because R_M may share some upper and lower limits with R. The rectangle R gets replaced by a new list of up to $2d + 1$ rectangles.

Given a set of rectangles $R_1, \dots, R_L$, the estimate of $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is

$$\hat{\mu}_{\text{divonne}} = \sum_{\ell=1}^{L} \frac{\mathbf{vol}(R_\ell)}{n} \sum_{i=1}^{n} f(\boldsymbol{X}_{\ell,i}) \tag{10.23}$$

where $\boldsymbol{X}_{\ell,i} \sim \mathbf{U}(R_\ell)$ independently. An alternative version of Divonne replaces random $\boldsymbol{X}_{\ell,i}$ by quasi-random points using Korobov rule. That is a lattice based QMC rule (see Chapter 16). The algorithm terminates when the Monte Carlo standard error is below a tolerance. If QMC points are used then they replace the standard error by

$$\frac{1}{2n}\sqrt{\sum_{\ell=1}^{L} s(R_\ell)^2}.$$

This estimate is $O(1/n)$ to reflect the improved convergence in QMC versus MC. They report that this error estimate is usually conservative.

Divonne is not well suited to unbounded integrands. When f is unbounded we may be able to use some problem-dependent reformulations to replace it by a bounded function with a known relationship to μ. In some cases, subtracting off a singular function will work. In others, a variable transformation may replace f by a bounded function.

Miser

Miser was developed in Press and Farrar (1990) and is also described in the Numerical Recipes books, such as Press et al. (2007, Chapter 7.9). That series of books includes code for Miser in multiple programming languages. Miser estimates the average of $f(\boldsymbol{x})$ over $\boldsymbol{x} \in [0,1)^d$ by recursively splitting the cube into two subcubes and estimating the average within each of those parts. It allocates a portion p (default 10%) of its budget of n function evaluations to deciding on which of the d axes the cube should be split. The rest of the budgeted evaluations are used to estimate $\int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ over each of the two subcubes.

The algorithm is recursive, treating those two subcubes the same way that the original one was, and stopping to ensure at least a minimum number (default 15) of function evaluations in every terminal subcube.

To describe the search for the split, suppose that we are working with the subcube $C = [\boldsymbol{a}, \boldsymbol{b}) = \prod_{j=1}^{d} [a_j, b_j) \subseteq [0,1)^d$. Let $c_j = (a_j + b_j)/2$ be the midpoint of the interval $[a_j, b_j)$. We can split C into left and right halves

$$\begin{aligned} C_{Lj} &= \{\boldsymbol{x} \in C \mid a_j \leqslant x_j < c_j\}, \quad \text{and} \\ C_{Rj} &= \{\boldsymbol{x} \in C \mid c_j \leqslant x_j < b_j\}, \end{aligned}$$

along the j'th axis.

Let $\mu_C = \int_C f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} / \mathbf{vol}(C)$ and $\sigma_C^2 = \int_C (f(\boldsymbol{x}) - \mu_C)^2 \,\mathrm{d}\boldsymbol{x} / \mathbf{vol}(C)$ be the mean and variance of f within C, respectively. From pN points uniformly distributed in C we can form estimates $\hat\sigma^2_{C_{Lj}}$ and $\hat\sigma^2_{C_{Rj}}$ for any of those sets that get at least 2 of our Np observations.

We have $n = N(1-p)$ points to distribute between two strata, C_{Lj} and C_{Rj}. If we put $n\theta$ points in C_{Lj} and the other $n(1-\theta)$ points in C_{Rj} then we can estimate $\mu_{C_{Lj}}$ by the average $\hat\mu_{C_{Lj}}$ of the $n\theta$ points in C_{Lj} and similarly for C_{Rj}. Our estimate of μ_C is

$$\hat\mu_{C,j} = \frac{1}{2}\bigl(\hat\mu_{C_{Lj}} + \hat\mu_{C_{Rj}}\bigr).$$

The variance of this estimate under simple random sampling is

$$\mathrm{Var}(\hat\mu_{C,j}) = \frac{1}{4}\Bigl(\frac{\sigma^2_{C_{Lj}}}{n\theta} + \frac{\sigma^2_{C_{Rj}}}{n(1-\theta)}\Bigr). \tag{10.24}$$

We can use the optimal stratified sampling allocation given by (8.17) of §8.4 and take $\theta = \sigma_{C_{Lj}}/(\sigma_{C_{Lj}} + \sigma_{C_{Lj}})$. The result is that

$$\mathrm{Var}(\hat\mu_{C,j}) = \frac{1}{4n}\bigl(\sigma_{C_{Lj}} + \sigma_{C_{Rj}}\bigr)^2. \tag{10.25}$$

Equation (10.25) motivates a rule of splitting on the axis j that minimizes $\hat\sigma_{C_{Lj}} + \hat\sigma_{C_{Rj}}$. Miser does not split this way. The reason is that the averages over left and right sides of C will not be estimated by simple random sampling but instead by a recursive application of Miser. They anticipate achieving a variance more like

$$\mathrm{Var}(\hat\mu_{C,j}) = \frac{1}{4}\Bigl(\frac{\sigma^2_{C_{Lj}}}{(n\theta)^\alpha} + \frac{\sigma^2_{C_{Rj}}}{(n(1-\theta))^\alpha}\Bigr) \tag{10.26}$$

for some quantity $\alpha \geqslant 1$. Equation (10.26) motivates choosing

$$\theta = (\sigma^2_{C_{Lj}})^{1/(1+\alpha)} \Bigm/ \Bigl[(\sigma^2_{C_{Lj}})^{1/(1+\alpha)} + (\sigma^2_{C_{Lj}})^{1/(1+\alpha)}\Bigr]$$

which leads to variance

$$\mathrm{Var}(\hat\mu_{C,j}) = \frac{1}{4n^\alpha}\Bigl[\sigma^{2/(1+\alpha)}_{C_{Lj}} + \sigma^{2/(1+\alpha)}_{C_{Lj}}\Bigr]^{1+\alpha}. \tag{10.27}$$

They settle on $\alpha = 2$ after some experimentation.

The cube C is split along the axis j that minimizes an estimate of (10.27). Instead of using the usual sample estimates they replace $\sigma_{C_{L,j}}$ in (10.27) by

$$\widetilde{\sigma}_{C_{L,j}} = \widehat{\max}\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in C_{L,j}\} - \widehat{\min}\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in C_{L,j}\}$$

where $\widehat{\max}$ and $\widehat{\min}$ are sample maxima and minima. The analogous estimate is used for $\widetilde{\sigma}_{C_{R,j}}$. They report that these values are empirically more reliable than sample standard deviations.

The preliminary sampling using np points to search for the best split may in some cases leave one or more of $C_{L,j}$ or $C_{R,j}$ empty. In that case axis j is not eligible for splitting. In the (very unlikely) event that all d axes are ineligible this way, they split on axis j for a random $j \sim \mathbf{U}\{1, 2, \ldots, d\}$.

Miser can have a bad result when the function f is nearly symmetric about the point $(1/2, 1/2, \ldots, 1/2) \in [0, 1)^d$. It may use up all of its possible splits in an attempt to partition the unit cube into 2^d pieces to be sampled independently. In such cases a small set of only d off-center splits might have been able to isolate the mode for further sampling. For such problems they introduce a dither parameter δ (such as 0.1 when not 0) and the splits are proposed at $c_j \pm \delta(b_j - a_j)/2$ instead of at c_j. The formulas all have to be modified to account for unequal volumes in C_{Lj} and C_{Rj}. The choice between $+1$ and -1 is made at random with equal probability.

Split Vegas

As noted above, Miser can have difficulties with unimodal functions. Conversely, Vegas is well suited to unimodal functions, but has a problem with multimodal functions. To see that problem, suppose that $f(\boldsymbol{x}) \in [0, 1)^d$ has two equal modes, one at $(0.1, 0.1, \ldots, 0.1)$ and one at $(0.9, 0.9, \ldots, 0.9)$. We might then find that each of the estimated factors q_j of the importance distribution has two modes, one at 0.1 and one at 0.9. Then $q(\boldsymbol{x}) = \prod_{j=1}^d q_j(x_j)$ has 2^d modes of which only 2 correspond to modes of f. The next sample has on average $2^{-d}n$ observations in each of those modes. It might well get zero observations near one of the true modes. When this happens the algorithm can lock on to the other mode and only estimate half of the integral. Zhou (1998) gives an example of two Gaussian bumps in $d = 9$ dimensions where this happens. The true integral is very close to 1 and Vegas gives an answer near 0.5 with a tiny standard error.

The split Vegas algorithm (Zhou, 1998) combines recursive splitting of $[0, 1)^d$ with Vegas fits inside each of the resulting subcubes. We can think of it as a 'Vegas within Miser' algorithm. Within a cubic region it fits Vegas and then inspects the marginal sampling distributions there. If it detects a strongly bimodal distribution for one of the factors then it splits the cube on that factor and applies split Vegas separately to the two resulting subcubes. The bimodality detection is based on fitting a cubic smoothing spline to the density factor q_j. It then looks for a local minimum in smoothed q_j with steep slopes in both directions towards the surrounding local maxima. If there are multiple strong local minima then the strongest one is used to define a split.

Combining stages

Adaptive importance sampling schemes give rise to a sequence of estimates $\hat{\mu}^{(k)}$ at stage k and usually we have estimates $\widehat{\mathrm{Var}}(\mu^{(k)})$ of their accuracy. These estimates are not independent because the sampling scheme used in stage k depends on the values we saw in stages 1 through $k-1$. Here we look at ways to combine them into an estimate of μ, a way to estimate the variance of that combination, and a square root rule that provides an advantageous combination.

Let $\mathcal{X}^{(k)} = (\boldsymbol{X}_1^{(k)}, \dots, \boldsymbol{X}_{n_k}^{(k)})$ hold the set of values used at stage k. The standard importance sampling mean and variance estimates satisfy conditional unbiasedness,

$$\begin{aligned}\mathbb{E}\bigl(\hat{\mu}^{(k)} \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k-1)}\bigr) &= \mu, \quad \text{and}\\ \mathbb{E}\bigl(\widehat{\mathrm{Var}}(\hat{\mu}^{(k)}) \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k-1)}\bigr) &= \mathrm{Var}\bigl(\hat{\mu}^{(k)} \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k-1)}\bigr).\end{aligned}$$

We will assume that $\mathrm{Var}(\hat{\mu}^{(k)} \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k-1)}) < \infty$.

Our combination is a simple weighted average over the stages, defined by nonrandom values $\omega_k \geqslant 0$ for $k = 1, \dots, K$ satisfying $\sum_{k=1}^K \omega_k = 1$. The estimate is

$$\hat{\mu}_\omega = \sum_{k=1}^K \omega_k \hat{\mu}^{(k)}. \tag{10.28}$$

For instance, the update (10.12) has $\omega_K = 1$ and $\omega_k = 0$ for $k < K$ in numerator and denominator of the self-normalized estimate. By contrast, the update (10.13) has $\omega_k \propto n_k$. The Vegas pooling method (10.22) does not fit this pattern because it uses data determined random weights.

Using fixed weights we get

$$\mathbb{E}(\hat{\mu}_\omega) = \sum_{k=1}^K \omega_k \mathbb{E}(\hat{\mu}^{(k)}) = \sum_{k=1}^K \omega_k \mathbb{E}\bigl(\mathbb{E}(\hat{\mu}^{(k)}) \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k)}\bigr) = \mu.$$

Although the estimates $\hat{\mu}^{(k)}$ are statistically dependent, they are uncorrelated. To see this, suppose that $1 \leqslant k < k' \leqslant K$. Then

$$\begin{aligned}\mathrm{Cov}(\hat{\mu}^{(k)}, \hat{\mu}^{(k')}) &= \mathbb{E}\bigl(\mathbb{E}((\hat{\mu}^{(k)} - \mu)(\hat{\mu}^{(k')} - \mu) \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k)})\bigr)\\ &= \mathbb{E}\bigl((\hat{\mu}^{(k)} - \mu)\mathbb{E}(\hat{\mu}^{(k')} - \mu \,|\, \mathcal{X}^{(1)}, \dots, \mathcal{X}^{(k)})\bigr) = 0.\end{aligned}$$

Therefore

$$\mathrm{Var}(\hat{\mu}_\omega) = \sum_{k=1}^K \omega_k^2 \,\mathrm{Var}(\hat{\mu}^{(k)}). \tag{10.29}$$

If we plug in estimates of $\mathrm{Var}(\hat\mu^{(k)})$ we get an unbiased estimate of $\mathrm{Var}(\hat\mu_\omega)$:

$$\begin{aligned}\mathbb{E}\biggl(\sum_{k=1}^K \omega_k^2\,\widehat{\mathrm{Var}}(\hat\mu^{(k)})\biggr) &= \sum_{k=1}^K \omega_k^2\,\mathbb{E}\bigl(\widehat{\mathrm{Var}}(\hat\mu^{(k)})\bigr)\\ &= \sum_{k=1}^K \omega_k^2\,\mathbb{E}\bigl(\mathbb{E}\bigl(\widehat{\mathrm{Var}}(\hat\mu^{(k)})\,|\,\mathcal{X}^{(1)},\dots,\mathcal{X}^{(k-1)}\bigr)\bigr)\\ &= \sum_{k=1}^K \omega_k^2\,\mathbb{E}\bigl(\mathrm{Var}(\hat\mu^{(k)})\,|\,\mathcal{X}^{(1)},\dots,\mathcal{X}^{(k-1)}\bigr)\\ &= \mathrm{Var}(\hat\mu_\omega).\end{aligned}$$

The last step follows because

$$\begin{aligned}\mathrm{Var}(\hat\mu^{(k)}) &= \mathbb{E}(\mathrm{Var}(\hat\mu^{(k)}\,|\,\mathcal{X}^{(1)},\dots,\mathcal{X}^{(k-1)})) + \mathrm{Var}(\mathbb{E}(\hat\mu^{(k)}\,|\,\mathcal{X}^{(1)},\dots,\mathcal{X}^{(k-1)}))\\ &= \mathbb{E}(\mathrm{Var}(\hat\mu^{(k)}\,|\,\mathcal{X}^{(1)},\dots,\mathcal{X}^{(k-1)})).\end{aligned}$$

The optimal weights are $\omega_k \propto 1/\mathrm{Var}(\hat\mu^{(k)})$ but these values are unknown. Vegas uses $\omega_k \propto 1/\widehat{\mathrm{Var}}(\hat\mu^{(k)})$ (equation (10.22)), but such data dependent weights can bring a severe bias. Consider a function $f(\boldsymbol{x}) \geqslant 0$ that is mostly near 0 but has one very narrow spike. If $\mathcal{X}^{(k)}$ misses that spike then both $\hat\mu^{(k)}$ and $\widehat{\mathrm{Var}}(\hat\mu^{(k)})$ are likely to be small. The result would be small estimates $\hat\mu^{(k)}$ getting large weight and vice versa, biasing the resulting estimate towards zero.

A rule that takes $\omega_k \propto \sqrt{k}$ when all of the n_k are equal is nearly optimal in a variety of different circumstances. Consider steady progress over the iterations with

$$\mathrm{Var}(\hat\mu^{(k)}) = \frac{\sigma^2}{nk^{r_0}} \tag{10.30}$$

where $\sigma > 0$ and the rate parameter satisfies $0 \leqslant r_0 \leqslant 1$. The lower bound $r_0 = 0$ describes a pessimistic setting where the iterations bring no improvement. The upper bound $r_0 = 1$ models a setting where adaptive Monte Carlo is essentially as good as using quasi-Monte Carlo. That is an optimistic upper bound, though not optimistic enough to include exponential convergence. Suppose that we do not know the true value r_0 and take $\omega_k \propto k^{r_1}$ instead. Then if we use $r_1 = 1/2$ we are never far from the optimum as Theorem 10.4 shows.

Theorem 10.4. *For $k = 1,\dots,K$, let $\mathrm{Var}(\hat\mu^{(k)}) = \sigma^2 k^{-r_0}/n$. Let $\omega = (\omega_1,\dots,\omega_K)$ with $\omega_k \propto k^{-r_0}$ and summing to 1. Let $\widetilde\omega = (\widetilde\omega_1,\dots,\widetilde\omega_K)$ with $\widetilde\omega_k \propto k^{-r_1}$ and summing to 1. Then for estimates $\hat\mu_\omega$ and $\hat\mu_{\widetilde\omega}$ given by (10.28),*

$$\min_{0\leqslant r_1\leqslant 1}\ \sup_{1\leqslant K<\infty}\ \max_{0\leqslant r_0\leqslant 1} \frac{\mathrm{Var}(\hat\mu_{\widetilde\omega})}{\mathrm{Var}(\hat\mu_\omega)} = \frac{9}{8}$$

and this minimum is attained at $r_1 = 1/2$. That is for $\omega_* = (\omega_{*1},\dots,\omega_{*K})$ with $\omega_{*k} \propto k^{1/2}$ and summing to one,

$$\sup_{1\leqslant K<\infty}\ \max_{0\leqslant r_0\leqslant 1} \frac{\mathrm{Var}(\hat\mu_{\omega_*})}{\mathrm{Var}(\hat\mu_\omega)} = \frac{9}{8}.$$

Proof. This is Theorem 4 of Owen and Zhou (1999). □

From Theorem 10.4 we see that the square root rule has efficiency within 12.5% of the unknown optimal rule for any number K of stages and for any true rate $r_0 \in [0,1]$. Exercise 10.11 looks at the efficiency which results from using $r_1 = 0$ when $r_0 = 1$ and vice versa.

10.7 Generalized antithetic sampling

In §8.2 we looked at antithetic sampling as it is usually used with antithetic (i.e., opposite) values of $\boldsymbol{x}$. The original formulation of antithetic variables by Hammersley and co-authors in the 1950s is much more general.

It is not necessary for $\widetilde{\boldsymbol{x}}$ to be a reflection of $\boldsymbol{x}$ through the center point of $\mathcal{D}$. Antithetic sampling works with other mappings. It suffices to have $\widetilde{\boldsymbol{X}} \sim p$ whenever $\boldsymbol{X} \sim p$. For the method to be beneficial, we should have $f(\boldsymbol{x})$ taking antithetic (i.e., nearly opposite) values to $f(\widetilde{\boldsymbol{x}})$.

If $\boldsymbol{X} \sim \mathbf{U}(0,1)^2$ then we may take $\widetilde{\boldsymbol{x}} = (1-x_1, x_2)$, reflecting just one component. We still find that $\widetilde{\boldsymbol{X}}$ has the same distribution as $\boldsymbol{X}$. More generally for $\boldsymbol{X} \sim \mathbf{U}(0,1)^d$ we can reflect any subset of the d components. A good subset is one where changing those components brings an antithetic value in f.

Consider for example $\boldsymbol{X}$ generated from $\mathcal{N}(0, I_2)$ by the Box-Muller transformation: $X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$ and $X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$ for $\boldsymbol{U} \sim \mathbf{U}(0,1)^2$. The value U_1 determines $\|\boldsymbol{X}\|$ while U_2 determines the angle that $\boldsymbol{X}$ makes with the horizontal axis. If we think that large and small $\|\boldsymbol{X}\|$ lead to antithetic values of f then we could define $\widetilde{\boldsymbol{X}}$ by a Box-Muller transformation of $\widetilde{\boldsymbol{X}} = (1-U_1, U_2)$ instead of the usual choice $\widetilde{\boldsymbol{X}} = -\boldsymbol{X}$. Which of these choices is better depends on f.

When p is not symmetric but $\boldsymbol{X} = \psi(\boldsymbol{U})$ for a transformation ψ of $\boldsymbol{U} \in (0,1)^s$ from Chapter 5 then we may use $\widetilde{\boldsymbol{X}} = \psi(\widetilde{\boldsymbol{U}})$ with $\widetilde{\boldsymbol{U}} = 1 - \boldsymbol{U}$ (componentwise). For acceptance-rejection we may use antithetic sampling to balance the proposals. Alternatively, we might apply antithetic sampling to balance the acceptance decisions.

The method extends beyond reflections. For $\boldsymbol{X} \sim \mathbf{U}(0,1)$ we may take $\widetilde{\boldsymbol{X}} = \boldsymbol{X} + 1/2 \ \mathsf{mod}\ 1$, a rotation modulo 1. Consider $f(\boldsymbol{x}) = |\boldsymbol{x} - 1/2|$. Ordinary antithetic sampling is of no use on this function but using the rotation will give the exact answer with $n = 2$ function evaluations.

We can generalize further by taking more than two function values at a time. For $\boldsymbol{x} \in [0,1]^d$ it is natural to consider evaluating f at 2^d values at a time, each one corresponding to a reflection of a subset of components of $\boldsymbol{x}$. This will only be attractive for small d.

For $\boldsymbol{x} \in \mathcal{D}$ define k transformations $\tau_j(\boldsymbol{x})$ such that $\tau_j(\boldsymbol{X}) \sim p$ whenever $\boldsymbol{X} \sim p$. Typically $\tau_1(\boldsymbol{x}) = \boldsymbol{x}$. The generalized antithetic sampling estimate is

$$\hat{\mu}_{\text{gen-anti}} = \frac{1}{n}\sum_{i=1}^{n/k}\sum_{j=1}^{k} f(\tau_j(\boldsymbol{X}_i)), \quad \boldsymbol{X}_i \overset{\text{iid}}{\sim} p.$$

For example, with $d = 1$, we could use rotation sampling,

$$\hat{\mu}_{\mathrm{rot}(k)} = \frac{1}{n}\sum_{i=1}^{n/k}\sum_{j=1}^{k} f\bigl(\{X_i + (j-1)/k\}\bigr)$$

where $\{z\} = z - \lfloor z \rfloor$. In an extreme version, $k = n$, so that one single point is rotated through n shifts of size $1/n$, modulo 1.

Generalized antithetic sampling will pay off when the random variable $g(\boldsymbol{X}) = (1/k)\sum_{j=1}^{k} f(\tau_j(\boldsymbol{X}))$ has variance below σ^2/k (using the usual cost assumptions). For any f with $\sigma^2 < \infty$ we have

$$0 \leqslant \mathrm{Var}(\hat{\mu}_{\text{gen-anti}}) \leqslant \frac{k\sigma^2}{n}. \tag{10.31}$$

The upper bound arises when we have chosen poorly and find that $f(\tau_j(\boldsymbol{x})) = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$. The lower bound arises when $\sum_{j=1}^{k} f(\tau_j(\boldsymbol{x}))$ is constant on $\boldsymbol{x} \in \mathcal{D}$.

These generalizations of antithetic sampling are continuous space versions of cluster sampling, a method used in survey sampling. In cluster sampling we define k–tuples of points in $\mathcal{D}$ and sample all members from n/k randomly chosen k–tuples.

The formulation of antithetic sampling in Hammersley's work was even more general than cluster sampling, because it allowed for unequal weighting of the function values. See page 51 of the chapter end notes. The simplest versions of antithetic sampling with one or two reflections are the most used. Some versions of randomized quasi-Monte Carlo, the shifted lattice rules of Chapter 16, can be viewed as cluster sampling.

10.8 Control variates with antithetics and stratification

In §9.10 we combined control variates with importance sampling. We can also use control variates in conjunction with antithetic sampling, and stratification, including Latin hypercube sampling. In all of these cases the combination is likely to change the optimal control variate coefficient vector β. Different control variates are helpful for these combinations than we would find helpful in simple Monte Carlo. In general, these combinations make it harder to identify promising control variates for a given f.

First we consider control variates with antithetic sampling. The goal is to find $\mu = \mathbb{E}(f(\boldsymbol{X}))$ for $\boldsymbol{X} \sim p$ on $\mathcal{D} \subset \mathbb{R}^d$. To each $\boldsymbol{x}$ there is an antithetic counterpart $\widetilde{\boldsymbol{x}}$ with $\widetilde{\boldsymbol{X}} \sim p$. There is also a vector h for which $\mathbb{E}(h(\boldsymbol{X})) = \theta$ is known.

Taking $\boldsymbol{X}_i \overset{\text{iid}}{\sim} p$, the combined estimate of μ is

$$\hat{\mu}_{\mathrm{anti},\beta} = \frac{1}{n}\sum_{i=1}^{n/2} f(\boldsymbol{X}_i) + f(\widetilde{\boldsymbol{X}}_i) - \beta^{\mathsf{T}}(h(\boldsymbol{X}_i) + h(\widetilde{\boldsymbol{X}}_i) - 2\theta)$$

$$= \frac{2}{n}\sum_{i=1}^{n/2} f_{\mathrm{E}}(\boldsymbol{X}_i) - \beta^{\mathsf{T}}(h_{\mathrm{E}}(\boldsymbol{X}_i) - \theta), \tag{10.32}$$

where $f_{\mathrm{E}}(\boldsymbol{x}) = (f(\boldsymbol{x}) + f(\widetilde{\boldsymbol{x}}))/2$ and $h_{\mathrm{E}}(\boldsymbol{x}) = (h(\boldsymbol{x}) + h(\widetilde{\boldsymbol{x}}))/2$. From equation (10.32), we see that the combination of control variates with antithetic sampling is equivalent to sampling the even part of f, using a control variate taken from the even part of h.

Operationally, combining control variates with antithetic sampling is straightforward. We simply compute the even parts of f and h, and then use the control variates method on them. Antithetic sampling generally changes the optimal value of β, and hence how we should estimate it. In the antithetic sampling context we should replace (8.33) by

$$\hat\beta_{\mathrm{E}} = \biggl(\sum_{i=1}^{n/2}\bigl(h_{\mathrm{E}}(\boldsymbol{x}_i) - \bar H\bigr)\bigl(h_{\mathrm{E}}(\boldsymbol{x}_i) - \bar H\bigr)^{\mathsf{T}}\biggr)^{-1}\sum_{i=1}^{n/2}\bigl(h_{\mathrm{E}}(\boldsymbol{x}_i) - \bar H\bigr) f_{\mathrm{E}}(\boldsymbol{x}_i). \tag{10.33}$$

Using the estimate $\hat\beta$ from (8.33) devised for ordinary sampling instead of (10.33) will reduce the efficiency of the method.

Using a single control variate h in a regression estimator will reduce the variance of antithetic sampling by a factor of $1-\rho_{\mathrm{E}}^2$ where $\rho_{\mathrm{E}} = \mathrm{Corr}(f_{\mathrm{E}}(\boldsymbol{X}), h_{\mathrm{E}}(\boldsymbol{X}))$. For multiple control variates the reduction is by $1 - R_{\mathrm{E}}^2$, where R_{E}^2 is the proportion of variance of $f_{\mathrm{E}}(\boldsymbol{X})$ explained by a linear regression on $h_{\mathrm{E}}(\boldsymbol{X})$.

Antithetic sampling can completely change which control variates are effective. It is only correlation between the even parts of the problem functions that matters. In particular, if h is an odd function, then $h_{\mathrm{E}} = 0$ and it is of no value whatsoever in antithetic sampling.

There are two ways to combine control variates with stratification. Both are well known in the survey sampling literature. We can define control variates within strata, and replace the ordinary within stratum averages by control variate estimates. Of course this method requires that we know the within stratum averages of the control variates. Alternatively, we can define global control variates which span all strata. That is the method we look at here.

Combining the grid-based stratification of §10.1 with control variates provides another illustration of how the control variate method has to change in response to changed sampling. Suppose that f is continuously differentiable on $[0,1]^d$ and that we have continuously differentiable control variates $h_1,\dots,h_J$, with $\int h_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \theta_j$ known. Then we may estimate $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by

$$\hat\mu_{\mathrm{gs},\beta} = \frac{1}{m^d}\sum_{i=1}^{m^d} f(\boldsymbol{X}_i) - \sum_{j=1}^{J}\beta_j(h_j(\boldsymbol{X}_i) - \theta_j)$$

which has variance asymptotic to $\sigma^2_{\mathrm{gs},\beta}/(12n^{1+2/d})$ where

$$\sigma^2_{\mathrm{gs},\beta} = \int \|\nabla f(\boldsymbol{x}) - \beta^{\mathsf{T}}\nabla h(\boldsymbol{x})\|^2\,\mathrm{d}\boldsymbol{x}.$$

Here $\nabla f(\boldsymbol{x})$ is the gradient vector of f (a row vector) and $\nabla h(x)$ is the J by d Jacobian matrix whose j'th row is the gradient of h_j. The asymptotically optimal value of β is

$$\beta_{\mathrm{gs}} = \Bigl(\int \nabla h(\boldsymbol{x})(\nabla h(\boldsymbol{x}))^{\mathsf{T}}\,\mathrm{d}\boldsymbol{x}\Bigr)^{-1}\int \nabla h(\boldsymbol{x})(\nabla f(\boldsymbol{x}))^{\mathsf{T}}\,\mathrm{d}\boldsymbol{x}, \tag{10.34}$$

which we may estimate by

$$\hat\beta_{\mathrm{gs}} = \Biggl(\sum_{i=1}^n \nabla h(\boldsymbol{X}_i)(\nabla h(\boldsymbol{X}_i))^{\mathsf{T}}\Biggr)^{-1}\sum_{i=1}^n \nabla h(\boldsymbol{X}_i)(\nabla f(\boldsymbol{X}_i))^{\mathsf{T}},$$

assuming we can compute first order partial derivatives of f and h_j.

A control variate h that correlates with f is not necessarily very useful. We need a control variate h whose gradient ∇h is a good linear predictor of ∇f.

Now suppose that we combine Latin hypercube sampling with control variates. The estimator is

$$\hat\mu_{\mathrm{lhs},\beta} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{X}_i) - \beta^{\mathsf{T}}(h(\boldsymbol{X}_i) - \theta)$$

where $\boldsymbol{X}_i$ are a Latin hypercube sample. It is unbiased because individually, $\boldsymbol{X}_i \sim \mathbf{U}(0,1)^d$. Let us suppose that the additive parts of f and h are smooth enough that they make negligible contribution to the sampling error. Then

$$\mathrm{Var}(\hat\mu_{\mathrm{lhs},\beta}) \doteq \frac{1}{n}\int \bigl(e_f(\boldsymbol{x}) - \beta^{\mathsf{T}}e_h(\boldsymbol{x})\bigr)^2\,\mathrm{d}\boldsymbol{x}$$

where $e_f = f - f^{\mathrm{add}}$ and $e_h = h - h^{\mathrm{add}}$ are the non-additive parts of f and h respectively. A good control variate h is one whose non-additive part (a sum of interactions) correlates with the non-additive part of f. It can be very hard to identify such a variable. If f is nearly additive then very likely the similar h that come to mind do so because their additive parts resemble f's, but the non-additive parts may not be strongly related.

With Latin hypercube sampling, it is not easy to estimate β either. The most straightforward approach is to run R independent replicates with each one a Latin hypercube sample of size n/R. Then we form R averages $\bar f_r$ and $\bar h_r$ for $r = 1,\dots,R$ of f and h over these Latin hypercube samples. The estimates $\hat\mu_{\mathrm{lhs},\beta}$ and $\hat\beta$ are then the minimizers of $\sum_{r=1}^r\bigl(\bar f_r - \mu_{\mathrm{lhs},\beta} - \beta^{\mathsf{T}}(\bar h_r - \theta)\bigr)^2$.

10.9 Bridge, umbrella and path sampling

Importance sampling and some variations of it, can be used to estimate the normalizing constants in distributions. In the physical sciences, the free energy of a system can be expressed in terms of a normalizing constant. Statisticians compute Bayes factors through normalizing constants.

The context for Bayes factors is as follows. We have some data $\mathcal{D}$ which could be a simple matrix of numbers, or it could be something more complicated like a collection of images or genetic sequences. The data has the distribution $p_u(\mathcal{D}\,|\,\theta)$ for a parameter $\theta \in \Theta$. Usually p_u is unnormalized. We may have two hypotheses about θ. Under H_0, $\theta \sim p_0$ and under H_1, $\theta \sim p_1$. Our prior information specifies $\mathbb{P}(H_0)$ and $\mathbb{P}(H_1) = 1 - \mathbb{P}(H_0)$. From Bayes' rule we find that the posterior odds in favor of H_1 are

$$\frac{\mathbb{P}(H_1\,|\,\mathcal{D})}{\mathbb{P}(H_0\,|\,\mathcal{D})} = \frac{\mathbb{P}(H_1)}{\mathbb{P}(H_0)} \times \frac{\mathbb{P}(\mathcal{D}\,|\,H_1)}{\mathbb{P}(\mathcal{D}\,|\,H_0)}. \tag{10.35}$$

The second factor in (10.35) is the Bayes factor for H_1 versus H_0, denoted $B_{1,0}$. We can rewrite the Bayes factor as

$$B_{1,0} = \frac{\int_\Theta p_u(\mathcal{D}\,|\,\theta)p_1(\theta)\,\mathrm{d}\theta / \int_\Theta p_1(\theta)\,\mathrm{d}\theta}{\int_\Theta p_u(\mathcal{D}\,|\,\theta)p_0(\theta)\,\mathrm{d}\theta / \int_\Theta p_0(\theta)\,\mathrm{d}\theta}.$$

When we work with normalized prior distributions p_j, the Bayes factor simplifies to

$$B_{1,0} = \frac{\int_\Theta p_u(\mathcal{D}\,|\,\theta)p_1(\theta)\,\mathrm{d}\theta}{\int_\Theta p_u(\mathcal{D}\,|\,\theta)p_0(\theta)\,\mathrm{d}\theta}.$$

Now $B_{1,0}$ is a ratio of normalizing constants for distributions $p_u(\mathcal{D}\,|\,\cdot)p_j(\cdot)$ on Θ.

We will emphasize computing ratios like $B_{1,0}$. When we want a single normalizing constant, it is often very effective to estimate its ratio to another known normalizing constant.

Our methods for estimating normalizing constants are based on importance sampling. We will describe the problem as integration over $\boldsymbol{x}$ (not θ) to match the notation we used for importance sampling, and because normalizing constants are also used outside of Bayesian applications. The normalizing constant for a distribution is usually written as either Z or c. We will use Z. The two densities may be on spaces of different dimension and, in some cases, we cannot generate samples those distributions either. Such restrictions affect some, but not all, of the methods we consider. Ways to mitigate these problems are discussed on page 49.

Suppose that we have two distributions $p_0(\boldsymbol{x}) = \widetilde{p}_0(\boldsymbol{x})/Z_0$ and $p_1(\boldsymbol{x}) = \widetilde{p}_1(\boldsymbol{x})/Z_1$, where $\widetilde{p}_j$ can be computed at any $\boldsymbol{x}$ and $Z_j > 0$. We may estimate $\rho = Z_1/Z_0$ by applying importance sampling separately in numerator and denominator. For $j = 1, 2$, let q_j be a normalized density with $q_j(\boldsymbol{x}) > 0$ whenever $p_j(\boldsymbol{x}) > 0$. Then we may use

$$\hat\rho_{q_1,q_0} = \frac{(1/n_1)\sum_{i=1}^{n_1} \widetilde{p}_1(\boldsymbol{X}_{i,1})/q_1(\boldsymbol{X}_{i,1})}{(1/n_0)\sum_{i=1}^{n_0} \widetilde{p}_0(\boldsymbol{X}_{i,0})/q_0(\boldsymbol{X}_{i,0})} \tag{10.36}$$

where $\boldsymbol{X}_{i,1} \overset{\mathrm{iid}}{\sim} q_1$ independently of $\boldsymbol{X}_{i,0} \overset{\mathrm{iid}}{\sim} q_0$. The optimal q_j are in fact the p_j, so we might use (10.36) with normalized distributions close to the p_j and with heavier tails than the p_j.

It is advantageous to use (10.36) with $q_1 = q_0$. Let q be a density with $q(\boldsymbol{x}) > 0$ whenever $\max(p_1(\boldsymbol{x}), p_2(\boldsymbol{x})) > 0$. Then we can take

$$\hat{\rho}_q \equiv \hat{\rho}_{q,q} = \frac{\sum_{i=1}^n \widetilde{p}_1(\boldsymbol{X}_i)/q(\boldsymbol{X}_i)}{\sum_{i=1}^n \widetilde{p}_0(\boldsymbol{X}_i)/q(\boldsymbol{X}_i)} = \frac{\sum_{i=1}^n \widetilde{p}_1(\boldsymbol{X}_i)/\widetilde{q}(\boldsymbol{X}_i)}{\sum_{i=1}^n \widetilde{p}_0(\boldsymbol{X}_i)/\widetilde{q}(\boldsymbol{X}_i)}, \tag{10.37}$$

for $\boldsymbol{X}_i \overset{\text{iid}}{\sim} q$, where $\widetilde{q}$ is an unnormalized version of q. Equation (10.37) brings two advantages over (10.36). We only need a sample from one distribution, and that distribution can be unnormalized.

We get a further simplification when we take $q = p_0$ in (10.37). If $p_0(\boldsymbol{x}) > 0$ whenever $p_1(\boldsymbol{x}) > 0$ then we may use

$$\hat{\rho}_{p_0} = \frac{1}{n}\sum_{i=1}^n \frac{\widetilde{p}_1(\boldsymbol{X}_i)}{\widetilde{p}_0(\boldsymbol{X}_i)}, \quad \boldsymbol{X}_i \overset{\text{iid}}{\sim} p_0. \tag{10.38}$$

Of the three estimators (10.36), (10.37) and (10.38), only the third one (i.e., $\hat{\rho}_{p_0}$) is unbiased. The other two are ratio estimators in which the law of large numbers applies to numerator and denominator. The methods we consider for estimating $\rho = Z_1/Z_0$ are generalizations of these estimators.

Bridge sampling

The method of ***bridge sampling***, also known as the ***acceptance ratio*** method works through a density function that bridges the domains of p_0 and p_1. Let $\Omega_0 = \{\boldsymbol{x} \mid p_0(\boldsymbol{x}) > 0\}$ and $\Omega_1 = \{\boldsymbol{x} \mid p_1(\boldsymbol{x}) > 0\}$ and let $\alpha(\boldsymbol{x})$ be a function defined on $\Omega_0 \cap \Omega_1$ such that

$$B \equiv \int_{\Omega_0\cap\Omega_1} \alpha(\boldsymbol{x})\widetilde{p}_0(\boldsymbol{x})\widetilde{p}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

satisfies $0 < |B| < \infty$. Then

$$\frac{\mathbb{E}_{p_0}(\widetilde{p}_1(\boldsymbol{X})\alpha(\boldsymbol{X}))}{\mathbb{E}_{p_1}(\widetilde{p}_0(\boldsymbol{X})\alpha(\boldsymbol{X}))} = \frac{\int_{\Omega_0\cap\Omega_1} \widetilde{p}_1(\boldsymbol{x})p_0(\boldsymbol{x})\alpha(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int_{\Omega_0\cap\Omega_1} \widetilde{p}_0(\boldsymbol{x})p_1(\boldsymbol{x})\alpha(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}} = \frac{B/Z_0}{B/Z_1} = \rho, \tag{10.39}$$

motivating the estimator

$$\hat{\rho}_{\mathrm{Bri},\alpha} = \frac{\frac{1}{n_0}\sum_{i=1}^{n_0} \widetilde{p}_1(\boldsymbol{X}_{i,0})\alpha(\boldsymbol{X}_{i,0})}{\frac{1}{n_1}\sum_{i=1}^{n_1} \widetilde{p}_0(\boldsymbol{X}_{i,1})\alpha(\boldsymbol{X}_{i,1})} \tag{10.40}$$

where $\boldsymbol{X}_{i,j} \sim p_j$ are independent.

Most or perhaps even all proposed bridge estimators satisfy $\alpha(\boldsymbol{x}) \geqslant 0$. Then we can define an unnormalized density $\widetilde{p}_{1/2} = \alpha(\boldsymbol{x})\widetilde{p}_0(\boldsymbol{x})\widetilde{p}_1(\boldsymbol{x})$ and find that

$$\rho = \frac{\mathbb{E}_{p_0}(\widetilde{p}_{1/2}(\boldsymbol{X})/\widetilde{p}_0(\boldsymbol{X}))}{\mathbb{E}_{p_1}(\widetilde{p}_{1/2}(\boldsymbol{X})/\widetilde{p}_1(\boldsymbol{X}))}. \tag{10.41}$$

This density $\widetilde{p}_{1/2}$ is the bridging density and B is its normalizing constant. A good choice of $\widetilde{p}_{1/2}$ will overlap both $\widetilde{p}_0$ and $\widetilde{p}_1$, which might only overlap each other a little. Bridge sampling generalizes (10.38) because after taking $\widetilde{p}_{1/2}(\boldsymbol{x}) = \widetilde{p}_1(\boldsymbol{x})$, the left side of (10.41) simplifies to $\mathbb{E}_{p_0}(\widetilde{p}_1(\boldsymbol{X})/\widetilde{p}_0(\boldsymbol{X}))$.

One well-known choice is $\alpha(\boldsymbol{x}) = 1/(\widetilde{p}_0(\boldsymbol{x})\widetilde{p}_1(\boldsymbol{x}))$ which leads to a method called the harmonic rule. It has bridging density $\widetilde{p}_{1/2} = \mathbf{U}(\Omega_0 \cap \Omega_1)$. It only defined when $\Omega_0 \cap \Omega_1$ has finite volume. The harmonic rule is extremely unstable owing to its use of reciprocal densities when used in (10.40). It commonly has infinite variance and is mentioned here only to warn against its use.

The choice of α which asymptotically minimizes $\mathrm{Var}(\log(\hat\rho))$ is known. Let $n = n_0 + n_1$, $s_0 = n_0/n$ and $s_1 = n_1/n$. If n goes to infinity and both s_j remain bounded away from 0 in this limit, then

$$\alpha_{\mathrm{opt}}(\boldsymbol{x}) \propto \frac{1}{s_0 p_0(\boldsymbol{x}) + s_1 p_1(\boldsymbol{x})} \propto \frac{1}{\rho\, s_0 \widetilde{p}_0(\boldsymbol{x}) + s_1 \widetilde{p}_1(\boldsymbol{x})} \tag{10.42}$$

is best (Bennett, 1976). This choice is also asymptotically best for the relative mean square error $\mathbb{E}((\hat\rho_{\mathrm{Bri},\alpha}/\rho - 1)^2)$. The optimal bridging distribution is

$$p_{1/2}(\boldsymbol{x}) \propto \widetilde{p}_{1/2}(\boldsymbol{x}) = \frac{\widetilde{p}_0(\boldsymbol{x})\widetilde{p}_1(\boldsymbol{x})}{\rho\, s_0 \widetilde{p}_0(\boldsymbol{x}) + s_1 \widetilde{p}_1(\boldsymbol{x})}. \tag{10.43}$$

Figure 10.6 illustrates two unnormalized densities together with their optimal bridging density and some suboptimal ones. The densities are $p_0 = \mathcal{N}(0,1)$ and $p_1 = \mathrm{Gam}(9)/2$. The optimal bridge density concentrates on the region where p_0 and p_1 overlap.

As is often the case, the best method for estimating ρ depends on the unknown true ρ. A practical approach to bridge estimation is to alternate between plugging $\hat\rho$ into the formula for optimal α and using that α to estimate ρ. We start with any $\hat\rho_{\mathrm{Bri}}^{(0)} > 0$ and then update via

$$\hat\rho_{\mathrm{Bri}}^{(k+1)} \leftarrow \frac{\displaystyle\frac{1}{n_0}\sum_{i=1}^{n_0} \frac{\widetilde{p}_1(\boldsymbol{X}_{i,0})}{\hat\rho_{\mathrm{Bri}}^{(k)}\, s_0\widetilde{p}_0(\boldsymbol{X}_{i,0}) + s_1\widetilde{p}_1(\boldsymbol{X}_{i,0})}}{\displaystyle\frac{1}{n_1}\sum_{i=1}^{n_1} \frac{\widetilde{p}_0(\boldsymbol{X}_{i,1})}{\hat\rho_{\mathrm{Bri}}^{(k)}\, s_0\widetilde{p}_0(\boldsymbol{X}_{i,1}) + s_1\widetilde{p}_1(\boldsymbol{X}_{i,1})}}. \tag{10.44}$$

The values $\widetilde{p}_j(\boldsymbol{X}_{i,j})$ only need to be computed once, and so the iterations are very fast. If some of the $\widetilde{p}_j(\boldsymbol{X}_{i,1-j})$ overflow numerically, then the update

$$\hat\rho_{\mathrm{Bri}}^{(k+1)} \leftarrow \frac{\frac{1}{n_0}\sum_{i=1}^{n_0}\bigl(\hat\rho_{\mathrm{Bri}}^{(k)}\, s_0 \widetilde{r}_{01}(\boldsymbol{X}_{i,0}) + s_1\bigr)^{-1}}{\frac{1}{n_1}\sum_{i=1}^{n_1}\bigl(\hat\rho_{\mathrm{Bri}}^{(k)}\, s_0 + s_1 \widetilde{r}_{10}(\boldsymbol{X}_{i,1})\bigr)^{-1}} \tag{10.45}$$

will be more reliable, where $\widetilde{r}_{01}(\boldsymbol{x}) = \widetilde{p}_0(\boldsymbol{x})/\widetilde{p}_1(\boldsymbol{x})$ and $\widetilde{r}_{10}(\boldsymbol{x}) = \widetilde{p}_1(\boldsymbol{x})/\widetilde{p}_0(\boldsymbol{x})$.

Theorem 10.5. *Let $\boldsymbol{X}_{i,j} \sim p_j$ be independent for $i = 1, \dots, n_j$ and $j = 1, 2$ and suppose that $\Omega_1 = \Omega_2 = \Omega$. Then $\hat\rho_{\mathrm{Bri}}^{(k)}$ defined by* (10.44) *converges to a unique*

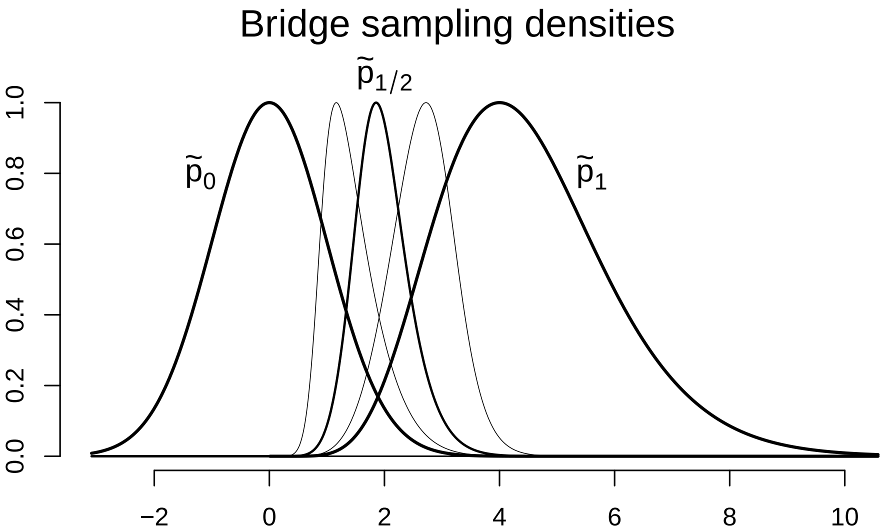

Figure 10.6: Unnormalized densities $\widetilde{p}_0$ and $\widetilde{p}_1$ are shown as thick lines. The optimal bridge density $\widetilde{p}_{1/2}$ in (10.43) is given as the dark curve in the middle. This $\widetilde{p}_{1/2}$ depends on the true unknown ρ. Suboptimal versions of $\widetilde{p}_{1/2}$ using 100ρ and $\rho/100$ are to the left and right of $\widetilde{p}_{1/2}$ (respectively) in thin lines.

limit $\hat{\rho}_{\mathrm{Bri}}$ as $k \to \infty$, the progress is governed by $|\hat{\rho}^{(k+1)}_{\mathrm{Bri}} - \hat{\rho}_{\mathrm{Bri}}| < |\hat{\rho}^{(k)}_{\mathrm{Bri}} - \hat{\rho}_{\mathrm{Bri}}|$ whenever $\hat{\rho}^{(k)}_{\mathrm{Bri}} \neq \hat{\rho}_{\mathrm{Bri}}$, and

$$\mathbb{E}\biggl(\Bigl(\frac{\hat{\rho}_{\mathrm{Bri}}}{\rho} - 1\Bigr)^2\biggr) \doteq \frac{1}{n}\biggl[\int_\Omega \Bigl(\frac{1}{s_0\widetilde{p}_0(\boldsymbol{x})} + \frac{1}{s_1\widetilde{p}_1(\boldsymbol{x})}\Bigr)\,\mathrm{d}\boldsymbol{x}\biggr]^{-1} - \frac{1}{n_0} - \frac{1}{n_1}. \quad (10.46)$$

Proof. If $\Omega_1 = \Omega_2$ then the ratios $l_{ji} = \widetilde{p}_1(\boldsymbol{x}_{i,j})/\widetilde{p}_0(\boldsymbol{x}_{i,j})$ satisfy $0 < l_{ji} < \infty$ and Theorem 2 of Meng and Wong (1996) applies. □

The significance of the right hand side of (10.46) is that this value is essentially the best possible relative error (Bennett, 1976; Meng and Wong, 1996) for a bridge estimator. As a result, the performance of $\hat{\rho}_{\mathrm{Bri}}$ obtained by (10.44) is, for large n_j, essentially as good as if we had used the unknown optimal α.

Umbrella sampling

An alternative to bridge sampling is the method of ***umbrella sampling***. In umbrella sampling, we work with (10.37), a ratio of two importance sampling estimators sampling from a common distribution q. The importance sampling density q which asymptotically minimizes the relative mean squared error is

$$q_{\mathrm{Umb}}(\boldsymbol{x}) \propto |p_1(\boldsymbol{x}) - p_0(\boldsymbol{x})| \propto |\rho\widetilde{p}_1(\boldsymbol{x}) - \widetilde{p}_0(\boldsymbol{x})| \quad (10.47)$$

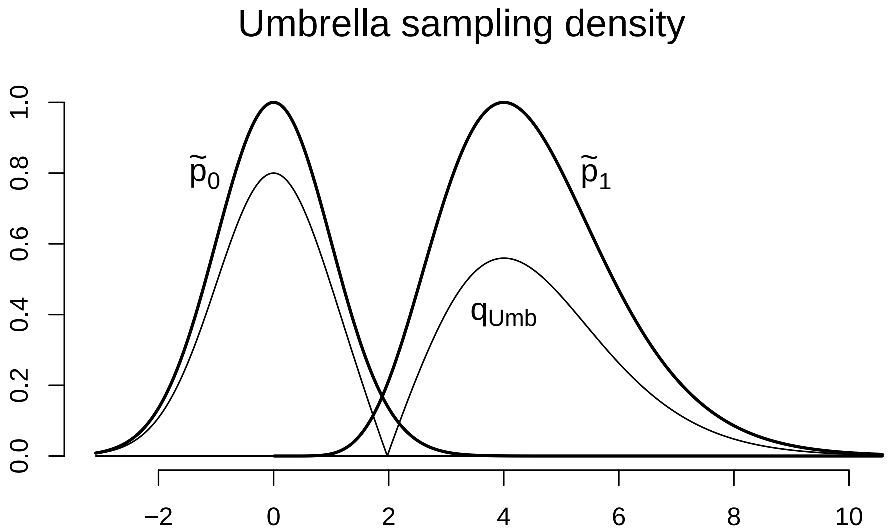

Figure 10.7: Unnormalized densities $\widetilde{p}_0$ and $\widetilde{p}_1$ are shown as thick lines. The optimal umbrella density q_{Umb} in (10.47) is given (unnormalized) as a thinner curve.

(Chen et al., 2000). Here

$$q_{\mathrm{Umb}}(\boldsymbol{x}) = \frac{|\rho\widetilde{p}_1(\boldsymbol{x}) - \widetilde{p}_0(\boldsymbol{x})|}{\int |\rho\widetilde{p}_1(\boldsymbol{x}) - \widetilde{p}_0(\boldsymbol{x})| \,\mathrm{d}\boldsymbol{x}}.$$

Like bridge sampling, the most efficient sampler requires knowledge of the true ρ. Umbrella sampling with q_{Umb} has a smaller asymptotic relative mean squared error than does bridge sampling with the optimal $\widetilde{p}_{1/2}$ (Chen et al., 2000).

Figure 10.7 shows the best umbrella density for the same densities shown in Figure 10.6. While the bridge sampling density concentrates in the region where p_0 and p_1 overlap, the umbrella density avoids regions where $p_0 = p_1$ which may well be where they overlap. If we use 100ρ or $\rho/100$ in (10.47) then q_{Umb} becomes almost identical to p_0 or p_1, respectively, in that example.

Though umbrella sampling attains a better optimum than bridge sampling, the form of q_{Umb} in (10.47) is hard to sample from after plugging in a guess for ρ. Also, to do so would require sampling new $\boldsymbol{X}_i$ whereas the bridge sampling estimator (10.44) does not need to regenerate points or recompute any $\widetilde{p}_j(\boldsymbol{x})$. Umbrella sampling appears to be more sensitive to a bad guess for ρ than bridge sampling is. As a result, the optimal q_{Umb} is harder to approach than the optimal $\widetilde{p}_{1/2}$ is. In practice, we would use a density q that covers both p_0 and p_1 (like an umbrella) and not one that vanishes on $\{\boldsymbol{x} \mid p_0(\boldsymbol{x}) = p_1(\boldsymbol{x})\}$.

Umbrella sampling has one strong advantage. It will work when there is no overlap between p_0 and p_1. Bridge sampling cannot handle that case. Because the bridge sampling iteration (10.44) is so effective, it is desirable to extend bridge sampling to distributions with little or no overlap. Path sampling and warping do just that.

Path sampling

We can extend bridge sampling to non-overlapping distributions by using some intermediate overlapping distributions. We form a set of distributions $p_\ell(\boldsymbol{x}) = \widetilde{p}_\ell(\boldsymbol{x})/Z_\ell$ for $\ell = 0, 1, \dots, L$, where p_0 is as before but p_L is our old p_1. We are interested in Z_L/Z_0 and we assume that $\Omega_\ell \cap \Omega_{\ell-1} \neq \varnothing$ for $1 \leqslant \ell \leqslant L$ where $\Omega_\ell = \{\boldsymbol{x} \mid p_\ell(\boldsymbol{x}) > 0\}$. Now $Z_L/Z_0 = \prod_{\ell=1}^L \rho_\ell$ where $\rho_\ell = Z_L/Z_{L-1}$. We can define distributions $\widetilde{p}_{\ell+1/2}$ to bridge $\widetilde{p}_\ell$ with $\widetilde{p}_{\ell+1}$ and then estimate

$$\frac{Z_L}{Z_0} = \prod_{\ell=1}^L \frac{\mathbb{E}_{p_\ell}(\widetilde{p}_{\ell-1/2}(\boldsymbol{X})/\widetilde{p}_\ell(\boldsymbol{X}))}{\mathbb{E}_{p_{\ell-1}}(\widetilde{p}_{\ell-1/2}(\boldsymbol{X})/\widetilde{p}_{\ell-1}(\boldsymbol{X}))} \tag{10.48}$$

by a corresponding product of L bridge sampling estimates.

This estimate puts $L-1$ stepping stones in between the first and last distribution and uses L bridges. If we let $L \to \infty$, we obtain the method of ***path sampling***, as described next.

We start with a parametric family of distributions $p(\cdot\,;\lambda) = \widetilde{p}(\cdot\,;\lambda)/Z_\lambda$ where $0 \leqslant \lambda \leqslant 1$ and $Z_\lambda > 0$ are normalizing constants. Here we connect the dots informally. For full details consult Gelman and Meng (1998). Now for a large integer $L \geqslant 1$,

$$\begin{aligned}
\log\left(\frac{Z_1}{Z_0}\right) &= \sum_{\ell=1}^L \log\left(\frac{Z_{\ell/L}}{Z_{(\ell-1)/L}}\right) = \sum_{\ell=1}^L \log\left(1 + \frac{Z_{\ell/L} - Z_{(\ell-1)/L}}{Z_{(\ell-1)/L}}\right)\\
&\doteq \sum_{\ell=1}^L \frac{Z_{\ell/L} - Z_{(\ell-1)/L}}{Z_{(\ell-1)/L}}\\
&\doteq \frac{1}{L}\sum_{\ell=1}^L \frac{d}{d\lambda}\log(Z_\lambda)\,\Big|_{\lambda=(\ell-1)/L}\\
&\doteq \int_0^1 \frac{d}{d\lambda}\log(Z_\lambda)\,\mathrm{d}\lambda.
\end{aligned}$$

Now we can sample $\lambda \sim \mathbf{U}(0,1)$ and average the derivative of $\log(Z_\lambda)$ to approximate $\log(Z_1/Z_0)$. For technical reasons, these continuous paths require all distributions in the path to have the same support set. It remains a big improvement over bridge sampling when p_0 and p_1 have small overlap, but if p_0 and p_1 don't overlap, then it is safter to use (10.48) for finite L.

Warping

Bridge sampling is impossible when p_0 and p_1 don't overlap, and is generally more accurate when the densities p_0 and p_1 overlap a lot. We will quantify the

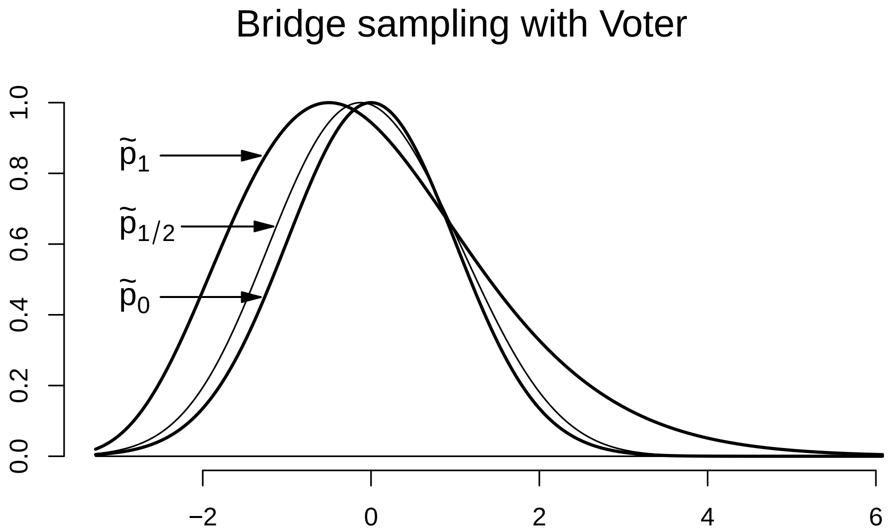

Figure 10.8: This figure shows Voter's method. The distribution $\widetilde{p}_1$ from Figure 10.6 has been shifted left to overlap $\widetilde{p}_0$. The new optimal bridge density $\widetilde{p}_{1/2}$ in (10.43) is also shown.

benefit of overlap below. Methods to bring p_0 and p_1 closer together can be viewed as warping those densities, or even, warping the space between them.

One simple method for obtaining a greater overlap in bridge sampling is ***Voter's method***, taking its name from Voter (1985). Let $p_0 = \widetilde{p}/Z_0$ and $p_1' = \widetilde{p}'/Z_1'$ be two given distributions of a random vector $\boldsymbol{X} \in \mathbb{R}^d$. If we sample $\boldsymbol{X} \sim p_1'$ then $\boldsymbol{X} - \Delta$ has a new distribution, call it p_1, with the same normalizing constant $Z_1 = Z_1'$. A judicious choice of Δ can make p_1 overlap p_0 more closely. Then we may apply bridge sampling to p_0 and p_1. Note that $p_1(\boldsymbol{x}) = p_1'(\boldsymbol{x}+\Delta)$.

Figure 10.8 shows an example where $\Delta = \mathbb{E}_{p_1'}(\boldsymbol{X}) - \mathbb{E}_{p_0}(\boldsymbol{X})$ which results in p_0 and p_1 having the same mean. The original method aligned modes of the distribution. Modes are easier to find in some applications. Incorporating Voter's method into the iteration (10.44) yields

$$\hat{\rho}_{\mathrm{Vot}}^{(k+1)} \leftarrow \frac{\dfrac{1}{n_0}\displaystyle\sum_{i=1}^{n_0} \frac{\widetilde{p}_1'(\boldsymbol{X}_{i,0}+\Delta)}{\hat{\rho}_{\mathrm{Vot}}^{(k)} s_0 \widetilde{p}_0(\boldsymbol{X}_{i,0}) + s_1 \widetilde{p}_1'(\boldsymbol{X}_{i,0}+\Delta)}}{\dfrac{1}{n_1}\displaystyle\sum_{i=1}^{n_1} \frac{\widetilde{p}_0(\boldsymbol{X}_{i,1}-\Delta)}{\hat{\rho}_{\mathrm{Vot}}^{(k)} s_0 \widetilde{p}_0(\boldsymbol{X}_{i,1}-\Delta) + s_1 \widetilde{p}_1'(\boldsymbol{X}_{i,1})}} \tag{10.49}$$

where $\boldsymbol{X}_{i,1}$ were drawn from (the original) p_1' and $\boldsymbol{X}_{i,0}$ were drawn from p_0.

The optimal shift Δ is not necessarily the difference in means or in modes. Those simple guesses are frequently close to optimal in simulated settings where the optimum can be found.

In addition to shifting the data, there are other operations that may make the distributions more similar, such as rotation or scaling. We will need to account for the Jacobians of those transformations. Before describing those methods, we study the relationship between overlap of p_0 and p_1, and effectiveness of bridge sampling.

One way to measure the overlap of two densities is via the Hellinger integral

$$\mathrm{HI} = \mathrm{HI}(p_0, p_1) = \int \sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}.$$

Large values of the Hellinger integral correspond to small values of the Hellinger distance

$$H(p_0,p_1) = \biggl(\int \Bigl(\sqrt{p_0(\boldsymbol{x})} - \sqrt{p_1(\boldsymbol{x})}\Bigr)^2 \,\mathrm{d}\boldsymbol{x}\biggr)^{1/2} = \sqrt{2 - 2\mathrm{HI}(p_0,p_1)}.$$

These integrals are over $\Omega_0 \cup \Omega_1$ with $p_j(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \notin \Omega_j$.

We can bound the optimal bridge sampling variance for two densities using the Hellinger integral. Meng and Schilling (2002) give this formula:

$$\frac{n_0+n_1}{n_0n_1}\biggl(\frac{2\sqrt{n_0n_1}}{(n_0+n_1)\mathrm{HI}} - 1\biggr) \leqslant \mathrm{Var}\bigl(\log(\hat\rho_{\mathrm{Bri},\alpha_{\mathrm{opt}}})\bigr) \leqslant \frac{n_0+n_1}{n_0n_1}\biggl(\frac{1}{\mathrm{HI}^2} - 1\biggr)$$

which for $n_0 = n_1 = n/2$ simplifies to

$$\frac{4}{n}\biggl(\frac{1}{\mathrm{HI}} - 1\biggr) \leqslant \mathrm{Var}\bigl(\log(\hat\rho_{\mathrm{Bri},\alpha_{\mathrm{opt}}})\bigr) \leqslant \frac{4}{n}\biggl(\frac{1}{\mathrm{HI}^2} - 1\biggr).$$

The Hellinger integral satisfies $0 \leqslant \mathrm{HI} \leqslant 1$. It is small values near 0 that cause us difficulty. If p_0 and p_1 overlap sufficiently near their peaks, then it does not matter that one might have tails much lighter than the other.

By applying a transformations to the data from p_1 we may be able to increase the amount of overlap with p_0. It turns out to be simpler to transform samples from both distributions to some common reference. That is also advantageous when there are more than two distributions to consider.

Let $\boldsymbol{X}$ have a continuous distribution with density $p(\boldsymbol{x}) = \widetilde{p}(\boldsymbol{x})/Z$. Let $\boldsymbol{Y} = T(\boldsymbol{X})$ be a smooth and invertible transformation of $\boldsymbol{X}$. Then the probability density function of $\boldsymbol{Y}$ is $q(\boldsymbol{y}) = p(T^{-1}(\boldsymbol{y}))|J(\boldsymbol{y})|$ where J is the Jacobian determinant of T^{-1}. That is, J is the determinant of the $d \times d$ matrix with rs element $\partial x_r/\partial y_s$.

If $\boldsymbol{Y}_{i,j} = T_j(\boldsymbol{X}_{i,j})$, then $\boldsymbol{Y}_{i,j}$ has density function

$$q_j(\boldsymbol{y}) = p_j(T_j^{-1}(\boldsymbol{y}))|J_j(\boldsymbol{y})| = \widetilde{q}_j(\boldsymbol{y})/Z_j$$

where $\widetilde{q}_j(\boldsymbol{y}) = p_j(T_j^{-1}(\boldsymbol{y}))|J_j(\boldsymbol{y})|$ and J_j is the Jacobian determinant of T_j^{-1}. In this parametrization, $\widetilde{q}_0$ and $\widetilde{q}_1$ have the same normalizing constants as $\widetilde{p}_0$ and $\widetilde{p}_1$, hence the same ratio. We may therefore run bridge sampling on densities $\widetilde{q}_j$ to estimate $\rho = Z_1/Z_0$. The combination is called ***warp bridge sampling***.

If $\boldsymbol{X}_{i,j}$ have a discrete distribution, then we do not multiply by $|J_j(\boldsymbol{y})|$, or equivalently, we can simply deem $|J_j(\boldsymbol{y})|$ to be 1. Scaling transformations are more likely to be useful for continuous variables than discrete ones, so we emphasize the continuous case.

We can write the optimal bridge sampling function in terms of the original unnormalized densities. Any positive multiple of

$$\alpha(\boldsymbol{y};\rho)=\frac{1}{\rho s_0\widetilde{p}_0(T_0^{-1}(\boldsymbol{y}))|J_0(\boldsymbol{y})|+s_1\widetilde{p}_1(T_1^{-1}(\boldsymbol{y}))|J_1(\boldsymbol{y})|}$$

is optimal. Given a starting $\hat\rho^{(0)}>0$ the update is

$$\hat\rho^{(k+1)}\leftarrow\frac{(1/n_0)\sum_{i=1}^{n_0}\widetilde{p}_1(T_1^{-1}(\boldsymbol{Y}_{i,0}))|J_1(\boldsymbol{Y}_{i,0})|\alpha(\boldsymbol{Y}_{i,0};\hat\rho^{(k)})}{(1/n_1)\sum_{i=1}^{n_1}\widetilde{p}_0(T_0^{-1}(\boldsymbol{Y}_{i,1}))|J_0(\boldsymbol{Y}_{i,1})|\alpha(\boldsymbol{Y}_{i,1};\hat\rho^{(k)})},\quad \boldsymbol{Y}_{i,j}=T_j(\boldsymbol{X}_{i,j}).$$

The most important transformations take the form $\boldsymbol{Y}_{i,j}=S_j^{-1}(\boldsymbol{X}_{i,j}-\Delta_j)$ for non-singular scaling matrices $S_j\in\mathbb{R}^{d\times d}$ and shift vectors $\Delta_j\in\mathbb{R}^d$. The Jacobian of T_j^{-1} is $|S_j|$. We assume that $|S_j|>0$ for the S_j we have chosen. We sample $\boldsymbol{X}_{i,j}\sim p_j$ independently for $i=1,\dots,n_j$ and $j=0,1$. Then we transform to $\boldsymbol{Y}_{i,j}=S_j^{-1}(\boldsymbol{X}_{i,j}-\Delta_j)$, and iterate the following

$$\begin{aligned}\hat\rho^{(k+1)}&\leftarrow\frac{|S_1|}{|S_0|}\frac{(1/n_0)\sum_{i=1}^{n_0}\widetilde{p}_1(S_1\boldsymbol{Y}_{i,0}+\Delta_1)\alpha(\boldsymbol{Y}_{i,0};\hat\rho^{(k)})}{(1/n_1)\sum_{i=1}^{n_1}\widetilde{p}_0(S_0\boldsymbol{Y}_{i,1}+\Delta_0)\alpha(\boldsymbol{Y}_{i,1};\hat\rho^{(k)})},\quad\text{where}\\ \alpha(\boldsymbol{y};\rho)&=\frac{1}{\rho s_0\widetilde{p}_0(S_0\boldsymbol{y}+\Delta_0)|S_0|+s_1\widetilde{p}_1(S_1\boldsymbol{y}+\Delta_1)|S_1|},\end{aligned}\tag{10.50}$$

starting from $\hat\rho^{(0)}>0$. Algorithm (10.50) yields the ***warp-II*** estimator. Voter's version is then called ***warp-I***.

Candidates for Δ_j are the mean of p_j as well as the mode. Then $\boldsymbol{Y}_{i,j}$ have common mean (respectively mode) zero. The matrix S_j could be a matrix square root of $\mathrm{Var}_{p_j}(\boldsymbol{X})$. Or it could be a matrix square root of the Hessian matrix of $-\log p_j(\boldsymbol{x})$ at Δ_j. We can construct distributions p_0 and p_1 where these transformations actually reduce overlap, but they are designed for unimodal p_0 and p_1 and will often greatly increase overlap. Figure 10.9 illustrates the warp-II densities.

Though warp-II transformations give distributions common values of location and spread, the transformed distributions may have different skewness. We can symmetrize the distributions to match the skewness. In this case, the unnormalized density $(\widetilde{p}_j(\boldsymbol{y})+\widetilde{p}_j(-\boldsymbol{y}))/2$ has the same normalizing constant that $\widetilde{p}_j$ has. That operation yields the ***Warp-III*** estimator. Figure 10.10 shows the warp-III density for our running example. There is now very high overlap between the distributions.

To implement warp-III with the linear transformation we sample $\boldsymbol{X}_{i,j}\sim p_j$ independently, set $\boldsymbol{Y}_{i,j}=S_j^{-1}(\boldsymbol{X}_{i,j}-\Delta_j)$, define

$$\widetilde{q}_j(\boldsymbol{y})=\frac{|S_j|}{2}\bigl(\widetilde{p}_j(S_j\boldsymbol{y}+\Delta_j)+\widetilde{p}_j(-S_j\boldsymbol{y}+\Delta_j)\bigr)$$

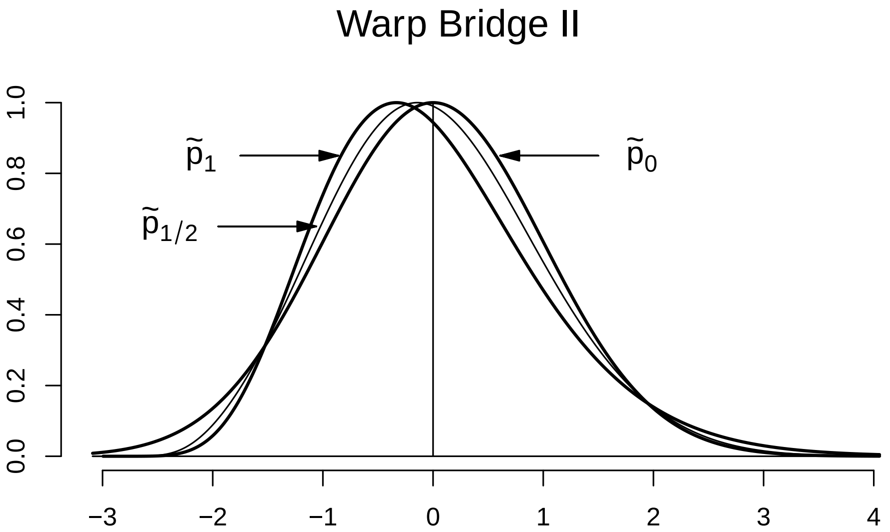

Figure 10.9: This figure shows the results of a warp-II transformation. The distribution $\widetilde{p}_1$ from Figure 10.6 has been linearly transformed to have mean 0 and variance 1 just like $\widetilde{p}_0$. The new optimal bridge density $\widetilde{p}_{1/2}$ in (10.43) is also shown.

and then iterate

$$\begin{aligned}\hat\rho^{(k+1)} &\leftarrow \frac{(1/n_0)\sum_{i=1}^{n_0}\widetilde{q}_1(\boldsymbol{Y}_{i,0})\alpha(\boldsymbol{Y}_{i,0};\hat\rho^{(k)})}{(1/n_1)\sum_{i=1}^{n_1}\widetilde{q}_0(\boldsymbol{Y}_{i,1})\alpha(\boldsymbol{Y}_{i,1};\hat\rho^{(k)})}, \quad \text{where}\\ \alpha(\boldsymbol{y};\rho) &= \frac{1}{\rho s_0\widetilde{q}_0(\boldsymbol{y})+s_1\widetilde{q}_1(\boldsymbol{y})},\end{aligned} \tag{10.51}$$

starting from $\hat\rho^{(0)} > 0$.

Table 10.2 shows results from four iterative algorithms to estimate the ratio $\rho = Z_1/Z_0$ of normalizing constants $Z_0 = \int_{-\infty}^{\infty}\exp(-x^2/2)\,\mathrm{d}x$ and $Z_1 = \int_0^{\infty}x^8\exp(-2x)\,\mathrm{d}x$. These are unnormalized $\mathcal{N}(0,1)$ and Gam(9)/2 distributions used in Figures 10.6 through 10.10. The answer is $\rho \doteq 31.417$. The methods used $n_0 = 1000$ observations from p_0 and $n_1 = 2000$ observations from p_1. Ordinarily one would use $n_0 = n_1$ unless the sampling costs were very different. Here, unequal sample sizes provide a better test for potential errors involving s_0 and s_1. The errors presented in Table 10.2 are based on $R = 1000$ independent replications.

Further issues

In the problems where normalization ratios are most needed, we may not be able to sample from either p_0 or p_1. Umbrella sampling does not require samples from p_0 or p_1, but bridge sampling does, and bridge sampling has a convenient

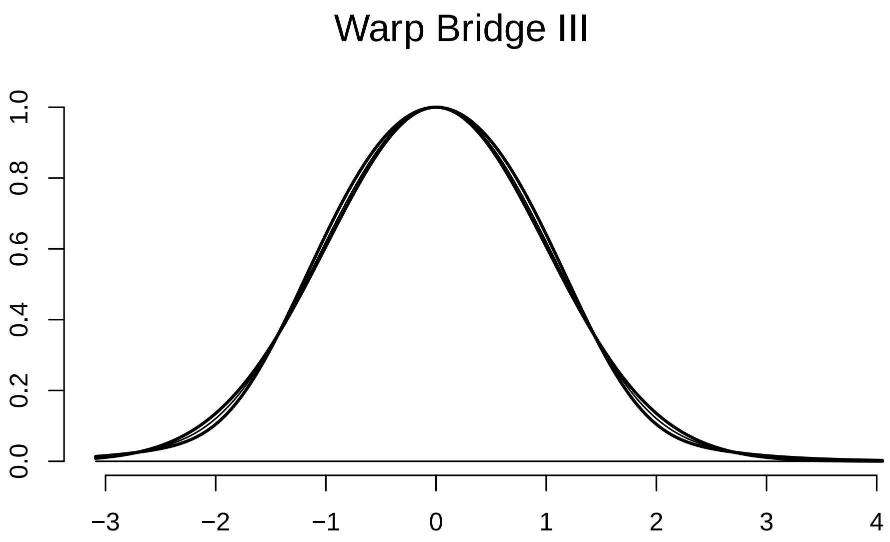

Figure 10.10: This figure shows the results of a warp-III transformation. The distribution $\widetilde{p}_1$ from Figure 10.6 has been linearly transformed to have mean 0 and variance 1 and then symmetrized. The new optimal bridge density $\widetilde{p}_{1/2}$ in (10.43) is also shown.

Method	Equation	RMSE/ρ
Bridge	(10.45)	0.1400
Voter	(10.49)	0.0094
Warp II	(10.50)	0.0053
Warp III	(10.51)	0.0018

Table 10.2: Relative root mean squared error for four estimates of ρ as described in the text. The bias contribution was negligible.

iteration. In practice, we may use Markov chain Monte Carlo (Chapter 11) to get samples that are approximately from p_0 and p_1. The sample points are not independent. It is then customary to adjust the values of s_0 and s_1 to account for the dependence. That is done by replacing n_j by $\widetilde{n}_j = n_j(1-\rho_j)/(1+\rho_j)$ where ρ_j is first-order autocorrelation from within chain j and then $s_j = \widetilde{n}_j/(\widetilde{n}_0+\widetilde{n}_1)$. The first-order autocorrelation depends on the quantity being averaged. Meng and Schilling (2002) used the median of a large set of the estimated first-order autocorrelations.

A second issue in normalization problems is that $\boldsymbol{X}_{i,0}$ and $\boldsymbol{X}_{i,1}$ may belong to spaces of different dimension, d_0 and d_1, respectively. Suppose that $d_1 > d_0$. One approach is to augment $\boldsymbol{X}_{i,0}$ with $d_1 - d_0$ further components, independent of the first d_0 components, from a distribution with normalizing constant 1.

Then we estimate the ratio of normalizing constants between augmented $\widetilde{p}_0$ and the original $\widetilde{p}_1$. Similar augmentations are used in reversible jump MCMC (Green, 1995). It can be difficult to construct effective augmenting variables. A second approach is to use warp bridge sampling to relate normalizing constants of each $\widetilde{p}_j$ to a standard distribution such as $\mathcal{N}(0, I_{d_j})$. Recall that bridge sampling is not adversely affected by light-tailed distributions such as $\mathcal{N}(0, I_d)$.

Chapter end notes

Hammersley and Morton (1956) consider a very general formulation of antithetic sampling that is now largely overlooked. They pay special attention to possibly unequal weightings on pairs of values. For $\boldsymbol{x} \in (0,1)$ and $0 < \alpha < 1$ the combinations $\alpha f(\alpha \boldsymbol{U}) + (1-\alpha) f(\alpha + (1-\alpha)\boldsymbol{U})$ and $\alpha f(\alpha \boldsymbol{U}) + (1-\alpha) f(1-(1-\alpha)\boldsymbol{U})$ are both unbiased estimates of $\mu = \int_0^1 f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ when $\boldsymbol{U} \sim \mathbf{U}(0,1)$. The optimal α for any arbitrary f is hard to find, but they report that the solution to $f(\alpha) = \alpha f(1) + (1-\alpha) f(1)$ often works well.

Hammersley and Mauldon (1956) pose the problem of optimal antithetic sampling, summing n different piecewise linear functions of a single $\boldsymbol{u} \sim \mathbf{U}(0,1)$ to estimate μ, but make little headway. For details on rotation sampling, see (Hammersley and Handscomb, 1964, Chapter 5) and Fishman and Huang (1983). Both papers find that rotation sampling is very well suited to periodic integrands. Randomized lattice rules, Chapter 16, are multidimensional generalizations of rotation sampling and they also are well suited to periodic integrands. In Chapter 16 we look at ways of replacing a given integrand by a periodic one having the same integral.

Grid-based stratification

Grid-based stratification and its $O(n^{-1/2-1/d})$ root mean square error were obtained by Dupach (1956), rediscovered by Haber (1966) and then combined with antithetic sampling by Haber (1967). Haber (1968) defines stochastic quadrature formulas wherein k random (not usually independent) points taken in the unit cube with possibly random weights give a generally unbiased integral estimate that is also exact for all polynomial integrands of degree r or lower, but not for all of degree $r+1$. Such rules, when applied within cubical strata, to functions that have continuous mixed partial derivatives of total order r, give an RMSE of $O(n^{-1/2-(r+1)/d})$. Ordinary stratification and antithetic sampling within strata correspond to $r=0$ with $k=1$ and $r=1$ with $k=2$, respectively. Haber (1969) proves that such rules exist, but for degree $r \geqslant 2$ the value of k required grows with d. He shows that for $r=2$ the minimal k is always at least $3d+1$. Some constructions for $r=2$ and even for $r=3$ appear in Haber (1969).

A	D	B	E	C
B	E	C	A	D
C	A	D	B	E
D	B	E	C	A
E	C	A	D	B

Table 10.3: A 5 by 5 Latin square.

Latin hypercubes and orthogonal arrays

The term 'Latin hypercube sampling' arises from a connection with Latin squares used in experimental design. Table 10.3 shows a 5 by 5 Latin square. Each row has exactly one of the five letters A, B, C, D and E. So does each column. If we were to extract the row and column coordinates corresponding to just one of the letters, such as A, those values suitably centered and scaled would comprise the points of a (centered) Latin hypercube sample for $d = 2$, given by (10.8).

The centered version (10.8) of Latin hypercube sampling was analyzed by Patterson (1954) who credits Yates for the scheme, and calls it lattice sampling. The application was to agricultural field trials.

McKay et al. (1979) is an early and influential paper on computer experiments. They introduced Latin hypercube sampling as a way to explore computationally the input space of a function, pointing out that it automatically stratifies on the important variables without the user having to know which those are. They prove that $\mathrm{Var}(\hat\mu_{\mathrm{LHS}}) \leqslant \mathrm{Var}(\hat\mu_{\mathrm{IID}})$ whenever the function being sampled is monotone in each of its d input variables.

Stein (1987) shows how the near additivity of the integrand plays a crucial role in the accuracy of LHS.

Shirley (1991) applies Latin hypercube sampling to ray tracing in computer graphics. He gave it the evocative name 'n-rooks'.

Background on orthogonal arrays may be found in Raghavarao (1971) or Hedayat et al. (1999). Owen (1992b) proposed randomized orthogonal array sampling for computer experiments and compared the quadrature accuracy to midpoint rules and Monte Carlo methods. Tang (1993) proposed orthogonal array based Latin hypercube sampling. Owen (1994) gave expressions for the variance of randomized orthogonal array sampling. To a good approximation they knock out the low order ANOVA components leaving an effective variance based on components of order $t + 1$ and larger.

Adaptive importance sampling

Marshall (1956) is an early reference for AIS. Berntsen et al. (1991) present an adaptive deterministic multidimensional quadrature based on recursive splitting.

For self-normalized AIS, Oh (1991) only considered cases where the update $\theta^{(k)} \leftarrow \widetilde{\mathbb{E}}^{(1:k)}(h)$ could be used. Oh and Berger (1993) give a stopping rule for AIS based on the width of the confidence interval for a target integrand. They also give conditions under which AIS converges. They require a uniform bound on the likelihood ratio. Evans (1991a) describes a chaining algorithm to use when the first AIS problem is too hard to solve. He replaces p by a density proportional to $p^{1/t}(\cdot)q(\cdot;\theta_1)^{1/u}$ for $(t,u) \in (0,\infty)^2$. Starting with $u=1$ and very large t, he solves a series of closely related problems defined by a decreasing sequence of t values. After t has decreased to 1 he starts raising u towards ∞. The value θ_1 is chosen to make the initial problem (large t and $u=1$) easy to do. This chaining process is similar to some of the temperature-based methods used in Markov chain Monte Carlo and simulated annealing. Further details on chaining are in Evans (1991b) which describes adaptive strategies for ordinary importance sampling.

Cornuet et al. (2012) use a different weighting than either Oh (1991) or Evans (1991a) used. Their adaptive multiple importance sampling (AMIS) algorithm takes the observations from the first k stages and weights them as if they had been sampled from the mixture density proportional to $q_u^{(1:k)}(\boldsymbol{x}) \equiv \sum_{s=1}^k n_k q^{(s)}(\boldsymbol{x};\theta^{(k)})$. They use self-normalized importance sampling estimating $\mathbb{E}_p(f(\boldsymbol{X}))$ by

$$\frac{\sum_{s=1}^k \sum_{i=1}^{n_k} p_u(\boldsymbol{X}_i^{(s)}) f(\boldsymbol{X}_i^{(s)})/q_u(\boldsymbol{X}_i^{(s)})}{\sum_{s=1}^k \sum_{i=1}^{n_k} p_u(\boldsymbol{X}_i^{(s)})/q_u(\boldsymbol{X}_i^{(s)})}.$$

The distribution of these estimates is difficult to study, because the data are not sampled independently from such a mixture. Instead, the later components are contructed based on data from the earlier ones.

The cross-entropy method is due to Rubinstein (1997). It is the subject of the book by Rubinstein and Kroese (2004) and there is a tutorial introduction in De Boer et al. (2005). The cross-entropy method is also used in some optimization problems.

The exponential convergence of adaptive importance sampling was observed empirically by Booth (1985) for some particle transport problems. Kollman (1993) proved exponential convergence under reasonable conditions but required samples to be run from every point in the state space. His argument has the nice feature of treating the sequence $\boldsymbol{\mu}^{(k)}$ itself as a Markov chain. A sample path started at $X_0 = i$ provides information about μ_i as well as μ_j if $X_n = j$ at any point along the trajectory. Kollman considers how to use that additional information.

Kollman's theory was generalized by Kollman et al. (1999) who allowed for fewer simulations to be done provided that the conditional means μ_i belong to a linear model $\boldsymbol{\mu} = X\beta$ for some $d \times k$ design matrix X and $\beta \in \mathbb{R}^k$. Baggerly et al. (2000) further relaxed the assumptions to allow continuous state spaces provided that there is a finite dimensional parametric model relating μ to each point in the space.

Some adaptive mixtures have been used for importance sampling. The following mixture of products of beta distributions

$$q(\boldsymbol{x}) = \sum_{m=1}^M \gamma_m \prod_{j=1}^d \frac{x_j^{\alpha_{mj}-1}(1-x_j)^{\beta_{mj}-1}}{\Gamma(\alpha_{mj})\Gamma(\beta_{mj})/\Gamma(\alpha_{mj}+\beta_{mj})}$$

was used by Zhou (1998). See also Owen and Zhou (1999). The context has $p = \mathbf{U}(0,1)^d$. That is $\boldsymbol{X}$ describes the random uniforms that get transformed into the problem inputs. The parameters satisfy $\gamma_m > 0$, $\sum_{m=1}^M \gamma_m = 1$, $\alpha_{mj} > 0$ and $\beta_{mj} > 0$. This factors in this family can produce modes in the middle of $(0,1)$ or singularities at either 0 or 1. For $f \geqslant 0$, the best q is a close approximation to $pf = f$. Zhou (1998) updates the parameters by minimizing an estimate of $\int (f(\boldsymbol{x}) - q(\boldsymbol{x}))^2 \,\mathrm{d}\boldsymbol{x}$.

Schürer (2004) develops quasi-Monte Carlo versions of both Vegas and MISER. He evaluates them using some of the test functions of Genz (1984). Pennanen and Koivu (2006) modify the partitioning strategy of MISER and attain better accuracy but also higher computing times in their examples. They avoid the symmetry problem not by dithering the breakpoint but by considering hyperrectangles into $L > 2$ congruent parts instead of the 2 used in MISER. The Vegas algorithm has a lot of strategies embedded in it. Chapter 3 in the dissertation Zhou (1998) gives a lengthy discussion of some of those ideas that are hard to discern from the article or code. The Appendix discusses convergence of the Vegas update.

Normalization

Bridge and umbrella sampling and related methods are covered thoroughly in Chen et al. (2000, Chapter 5). They provide proofs for the asymptotic optimality results quoted here. The primary techniques one needs are the delta method, the law of large numbers, and for optimality, the Cauchy-Schwarz inequality. To apply Cauchy-Schwarz we usually need to know what the optimal function is. In many cases that can be found by the calculus of variations (Gelfand and Fomin, 2000).

Bridge sampling was developed by Bennett (1976) for problems in physics and reinvented by Meng and Wong (1996) for uses in statistical inference. (It appears that the former paper was brought to the latter authors' attention by a reviewer.) The latter paper emphasizes applications to Bayes factors, missing data imputation and computation of likelihood ratios. They contributed the iterative plug-in estimator (10.44) and some more asymptotic analysis. The former paper considered using dependent samples from the two distributions, as one would use in Markov chain Monte Carlo (see Chapter 11). The optimal α function is usually not known when dependent observations are used. The problem of estimating ρ by bridge sampling has been studied by relating it to logistic regression. See Geyer (1994) and Shirts et al. (2003) for details of this approach.

The harmonic rule is due to Newton and Raftery (1994). The severe numerical instability of this method was pointed out in a discussion by Radford Neal, but the method still gets used.

Umbrella sampling was developed by Torrie and Valleau (1977). Chen et al. (2000) call it ratio importance sampling and present a statistical analysis.

Path sampling is described by Gelman and Meng (1998). They work out the optimal path between two distributions and relate it to a 'Rao Geodesic'.

For a toy example, with $p_0 = \mathcal{N}(0,1)$ and $p_1 = \mathcal{N}(D,1)$, three path sampling methods attain asymptotic variances that are quadratic, linear and logarithmic in D, while bridge methods and importance sampling are exponential in D.

Voter's (1985) method was generalized to warp bridge sampling in Meng and Schilling (2002). The warp bridge sampling described there is best suited for unimodal densities. For a multimodal density with a mixture representation, ideally one component per mode, it becomes possible to warp each mode separately to become similar to the $\mathcal{N}(0, I_d)$ distribution. That operation can greatly increase overlap. See Wang and Meng (2013).

Sinharay and Stern (2005) compare a number of methods for estimating normalizing constants for Bayesian generalized linear mixed models. They find particularly good results for warping the bridge estimator and for a method of that uses warping with plain importance sampling. Warp III was particularly accurate and the importance sampling version somewhat faster than the bridge sampling version.

Nonparametric likelihood

Kong et al. (2003) introduce a nonparametric likelihood approach to Monte Carlo. It unifies several of the importance sampling and normalization constant estimators. They use a nonparametric maximum likelihood (NPMLE) approach to combining observations from multiple distributions.

The NPMLE for sampling from multiple biasing distributions is due to Vardi (1985) with further analysis by Gill et al. (1988). The likelihood equations for estimating normalizing constants are also given in Geyer (1994). Tan (2004) gives several theorems on optimality of the NPMLE approach compared to multiple importance sampling and stratified importance sampling. The nonparametric likelihood can also be used in likelihood ratios to construct confidence intervals and hypothesis tests. The resulting empirical likelihood method is described in Owen (2001). Qin (1993) shows that empirical likelihood confidence intervals and tests apply in the multiple sample setting.

Kong et al. (2003) motivate the NPMLE of $\mu = \int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by supposing that $\mathrm{d}\boldsymbol{x}$ is the unknown, unlike a sampling approach in which p may appear to be the unknown given sample values $f(\boldsymbol{X}_i)$. Diaconis (1988) has an interesting discussion on what it means for a mathematical quantity to be unknown. He also shows that some familiar methods from numerical analysis can be viewed as Bayesian estimates.

Exercises

10.1. For equation (10.3) prove that $\mathbb{E}(\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{gs}})) = \mathrm{Var}(\hat\mu_{\mathrm{gs}})$.

10.2. We saw in §10.1 that stratification can achieve a better variance rate than $O(n^{-1})$. This could change how we measure efficiency.

Suppose that two stratification methods are being compared. The baseline method yields variance $A_0 n^{-r}$ for $r > 1$ and costs nc_0, when the sample size is n. The alternative method yields variance $A_1 n^{-r}$ and costs nc_1 under these circumstances. Of course A_0, A_1, c_0, and c_1 are all positive numbers.

a) Work out how to measure the relative efficiency of the new method to the old one. Give an expression to replace equation (8.1) for this setting.

b) Suppose that $r = 2$, $A_0 = c_0 = 1$ and $A_1 = 0.01$. How small must c_1 be in order for the new method to represent an efficiency improvement over the standard method?

10.3. An alternative variance estimate for grid-based stratification is

$$\widetilde{\mathrm{Var}}(\hat\mu_{\mathrm{gs}}) = (\hat\mu_{\mathrm{gs},1} - \hat\mu_{\mathrm{gs},2})^2/4$$

where in the notation of equation (10.3), $\hat\mu_{\mathrm{gs},1} = m^{-d}\sum_{i=1}^n f(\boldsymbol{X}_i)$ and $\hat\mu_{\mathrm{gs},2} = m^{-d}\sum_{i=1}^n f(\boldsymbol{X}_i')$. This estimate is unbiased, but not as good as $\widehat{\mathrm{Var}}(\mu_{\mathrm{gs}})$. It is tempting to use it, because it may be computed directly from $\hat\mu_{\mathrm{gs},\ell}$ for $\ell = 1, 2$. But as this exercise demonstrates, there is some loss of precision.

a) First prove that $\mathbb{E}(\widetilde{\mathrm{Var}}(\hat\mu_{\mathrm{gs}})) = \mathrm{Var}(\hat\mu_{\mathrm{gs}})$.

b) Now, for $d = 2$ and $m = 32$ and the test function $f(\boldsymbol{x}) = x_1^2\exp(x_1x_2)$ over $\boldsymbol{x} \in [0,1]^2$ (from Sloan and Joe (1994)) generate 1000 independent replicates of $\hat\mu_{\mathrm{gs}}$. For each replicate compute $\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{gs}})$ and $\widetilde{\mathrm{Var}}(\hat\mu_{\mathrm{gs}}))$. Compare the histograms of these two variance estimates. Using your sample data, what do you estimate $\mathrm{Var}(\widetilde{\mathrm{Var}}(\hat\mu_{\mathrm{gs}}))/\mathrm{Var}(\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{gs}}))$ to be? (Do not compute a confidence interval for this ratio of variances.)

10.4. For $\boldsymbol{x} \in [0,1]^{1000}$ let $g(\boldsymbol{x}) = 1 + \prod_{j=1}^{1000} x_j^\alpha/j^2$ and $f(\boldsymbol{x}) = 1/g(\boldsymbol{x})$.

a) Using $n = 1000$ and $\alpha = 1$, report a simple Monte Carlo estimate $\hat\mu$ of $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and also the usual estimate of $\mathrm{Var}(\hat\mu)$.

b) Compute independent Latin hypercube sampling estimates $\hat\mu_{\mathrm{LHS},r}$ of μ for $r = 1, \dots, 10$, using $n = 100$ and $\alpha = 1$ each time. Report their average $\hat\mu_{\mathrm{LHS}}$, along with an unbiased estimate of $\mathrm{Var}(\hat\mu_{\mathrm{LHS}})$.

c) Repeat the two previous parts with $\alpha = 1/2$.

This is the 1000-dimensional version of a function from Kuo et al. (2011).

10.5. For $x \in [0,1]$ define $g(x) = 1$ for $1/2 \leqslant x \leqslant 1$ and $g(x) = -1$ for $0 \leqslant x < 1/2$. Let $f(\boldsymbol{x}) = \prod_{j=1}^d g(x_j)$. This function is 1 if $\boldsymbol{x}$ has an even number of components in $[1/2, 1]$ and it is -1 for an odd number. The best additive approximation to f is the function that is 0 on $[0,1]^d$. As a result, Latin hypercube sampling should bring no benefit. A Monte Carlo sample of size n from f has mean 0 and variance $1/n$. Estimate $\mathrm{Var}(\hat\mu_{\mathrm{LHS}})$ for this function when $n = 1000$ and $d = 10$. Base your estimate on 1000 independent Latin hypercube samples. Form a 99% confidence interval for this variance. Hint: $\mathrm{Var}(\hat\mu_{\mathrm{LHS}}) = \mathbb{E}(\hat\mu_{\mathrm{LHS}}^2)$.

10.6 (Research)**.** Proposition 10.4 states that $\mathrm{Var}(\hat{\mu}_{\mathrm{LHS}}) \leqslant \sigma^2/(n-1)$ for $n \geqslant 1$. The proof in Owen (1997) gives the result as a corollary of more general formulas for scrambled net quadrature.

For this exercise, develop a direct proof of Proposition 10.4 without using theory of scrambled nets. You may use the result in Example 1 of Owen (1994). That result shows that $\mathrm{Var}(\hat{\mu}_{\mathrm{LHS}}) \leqslant \sigma^2/(n-1)$ holds for functions f that are constant within cubical cells of side $1/n$, that is, functions with $f(\boldsymbol{x}) = f(\lfloor n\boldsymbol{x} \rfloor/n)$. Extend this result to functions of finite variance that are not necessarily constant within cubical cells of side $1/n$.

10.7. For $\boldsymbol{x} \in \mathbb{R}^2$ let $f(\boldsymbol{x}) = \min(x_1, x_2)$. Use the cross-entropy method to estimate $\mu = \mathbb{P}(f(\boldsymbol{X}) > 6)$ where $\boldsymbol{X} \sim \mathcal{N}(0, I_2)$. We know that $\mu = \Phi(-6)^2 \doteq 9.734 \times 10^{-19}$. Run 10 iterations with $n = 10{,}000$.

a) Report $\theta^{(10)}$ and $\hat{\mu}^{(10)} - \mu$.

b) What was the first stage k, if any, that had the 99'th quantile of $f(\boldsymbol{X}_i^{(k)})$ above 6?

c) Repeat the computation 49 more times independently. Report the mean of all 50 estimates $\hat{\mu}^{(10)}$ and a 99% confidence interval for μ based on them.

d) Plot the 50 ending points $\theta^{(10)}$.

10.8. Repeat Exercise 10.7, but this time use $f(\boldsymbol{x}) = \max(x_1, x_2)$. We know that $\mu = 2\Phi(-6) - \Phi(-6)^2 \doteq 1.973 \times 10^{-9}$.

10.9. Let $q(\boldsymbol{X}; \lambda) = \prod_{j=1}^{d} e^{-\lambda_j} \lambda_j^{X_j}/X_j!$ for $\lambda = (\lambda_1, \dots, \lambda_d) \in (0, \infty)^d$.

a) Show that the natural exponential family parameter for q is $\theta = \log(\lambda)$ componentwise.

b) Show that the cross-entropy update (10.16) for this family is

$$\theta^{(k+1)} \leftarrow \log\biggl(\sum_{i=1}^{n_k} H_i^{(k)} \boldsymbol{X}_i^{(k)} \Bigm/ \sum_{i=1}^{n_k} H_i^{(k)}\biggr).$$

10.10. Implement the adaptive importance sampling scheme (10.18) using the safer update (10.19). Run it $N = 1000$ times with $K = 400$ steps. Call each run a success if an absolute error below 10^{-12} was attained in the first K steps.

a) What fraction of your runs were successes?

b) If you had any successful runs, then report the average value k in those runs, at which $|\hat{\mu}^{(k)} - \mu| < 10^{-12}$ first appeared.

c) What fraction of the N runs had any iteration with $\hat{\mu}^{(k)} < 1$? (Those runs would have failed without the safer update (10.19).)

10.11. In the setting of Theorem 10.4 let $K = 10$, $r_0 = 0$ and $r_1 = 1$.

a) What is $\mathrm{Var}(\hat{\mu}_{\widetilde{\omega}})/\mathrm{Var}(\hat{\mu}_{\omega})$?

b) If instead, $r_0 = 1$ and $r_1 = 0$, what is $\mathrm{Var}(\hat{\mu}_{\widetilde{\omega}})/\mathrm{Var}(\hat{\mu}_{\omega})$?

c) Answer both of the previous parts using $K = 100$ stages.

10.12. Let $\widetilde{p}_0(x) = \exp(-x^2/2)$ and $\widetilde{p}_1(x) = x^2(1-x)^3\mathbb{1}_{0<x<1}$ be two unnormalized densities on $\mathbb{R}$. Compare bridge, Voter, warp I and warp II sampling estimates of $\rho = Z_1/Z_0$. Use $n_0 = 1000$ and $n_1 = 1200$ and repeat 1000 times. Report the RMSE/ρ. First find the true ρ.

10.13 (Research)**.** Suppose that we know the marginal distribution of the first component X_1 of $\boldsymbol{X}$ under both p_0 and p_1. Devise a warp estimator of $\rho = Z_1/Z_0$ which takes advantage of this information.

Bibliography

Baggerly, K., Cox, D., and Picard, R. (2000). Exponential convergence of adaptive importance sampling for Markov chains. *Journal of applied probability*, 37(2):342–358.

Bennett, C. H. (1976). Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22:245–268.

Berntsen, J., Espelid, T. O., and Genz, A. (1991). Algorithm 698: DCUHRE: an adaptive multidemensional integration routine for a vector of integrals. *ACM Transactions on Mathematical Software (TOMS)*, 17(4):452–456.

Booth, T. E. (1985). Exponential convergence for Monte Carlo particle transport. *Transactions of the American Nuclear Society*, 50(CONF-851115-).

Bush, G. A. (1952). Orthogonal arrays of index unity. *Annals of Mathematical Statistics*, 23(3):426–434.

Chen, M.-H., Shao, Q.-M., and Ibrahim, J. G. (2000). *Monte Carlo methods in Bayesian computation*. Springer, New York.

Cornuet, J., Marin, J.-M., Mira, A., and Robert, C. P. (2012). Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812.

De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67.

Diaconis, P. (1988). Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics IV, in two volumes*, volume 1, pages 163–176.

Dupach, V. (1956). Stockasticke pocetni metodi. *Casopis pro pestování matematiky*, 81(1):55–68.

Evans, M. (1991a). Adaptive importance sampling and chaining. In Flournoy, N. and Tsutakawa, R. K., editors, *Statistical Numerical Integration, Contemporary Mathematics*, volume 115, pages 137–143. American Mathematical Society, Providence, RI.

Evans, M. (1991b). Chaining via annealing. *The Annals of Statistics*, pages 382–393.

Fishman, G. S. and Huang, B. D. (1983). Antithetic variates revisited. *Communications of the ACM*, 26(11):964–971.

Friedman, J. H. and Wright, M. H. (1979). User's guide for DIVONNE. Technical report, Stanford Linear Accelerator Center.

Friedman, J. H. and Wright, M. H. (1981). A nested partitioning procedure for numerical multiple integration. *ACM Transactions on Mathematical Software (TOMS)*, 7(1):76–92.

Gelfand, I. M. and Fomin, S. V. (2000). *Calculus of variations.* Dover, Mineola, NY.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185.

Genz, A. (1984). Testing multidimensional integration routines. In Ford, B., Rault, J. C., and Thomasset, F., editors, *Proceedings of international conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94. Elsevier North-Holland, Inc.

Geyer, C. J. (1994). Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo. Technical report, University of Minnesota.

Gill, R. D., Vardi, Y., and Wellner, J. A. (1988). Large sample theory of empirical distributions in biased sampling models. *The Annals of Statistics*, 16:1069–1112.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Haber, S. (1966). A modified Monte Carlo quadrature. *Mathematics of Computation*, 20:361–368.

Haber, S. (1967). A modified Monte Carlo quadrature, II. *Mathematics of Computation*, 21:388–397.

Haber, S. (1968). A combination of Monte Carlo and classical methods for evaluating multiple integrals. *Bulletin of the American Mathematical Society*, 74(4):683–686.

Haber, S. (1969). Stochastic quadrature formulas. *Mathematics and Computation*, 23(108):751–764.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo methods*. Methuen, London.

Hammersley, J. M. and Mauldon, J. G. (1956). General principles of antithetic variates. *Mathematical proceedings of the Cambridge philosophical society*, 52(3):476–481.

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.

Hedayat, A. S., Sloane, N. J. A., and Stufken, J. (1999). *Orthogonal Arrays: Theory and Applications*. Springer, New York.

Kollman, C. (1993). *Rare event simulation in radiation transport*. PhD thesis, Stanford University.

Kollman, C., Baggerly, K., Cox, D., and Picard, R. (1999). Adaptive importance sampling on discrete Markov chains. *Annals of Applied Probability*, pages 391–412.

Kong, A., McCullagh, P., Meng, X.-L., Nicolae, D., and Tan, Z. (2003). A theory of statistical models for Monte Carlo integration (with discussion). *Journal of the Royal Statistical Society, Series B*, 65(3):585–618.

Kuo, F. Y., Schwab, C., and Sloan, I. H. (2011). Quasi-Monte Carlo methods for high-dimensional integration: the standard (weighted Hilbert space) setting and beyond. *The ANZIAM Journal*, 53:1–37.

Lepage, G. P. (1978). A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192–203.

Lepage, G. P. (1980). VEGAS - an adaptive multi-dimensional integration program. Technical Report CLNS-80/447, Cornell University.

Loh, W.-L. (1996). On Latin hypercube sampling. *Annals of Statistics*, 24(5):2058–2080.

Marshall, A. W. (1956). The use of multi-stage sampling schemes in Monte Carlo computations. In Meyer, M. A., editor, *Symposium on Monte Carlo methods*, pages 123–40, New York. Wiley.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586.

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, 6:831–860.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society, Series B*, pages 3–48.

Oh, M.-S. (1991). Monte Carlo integration via importance sampling: Dimensionality effect and an adaptive algorithm. In Flournoy, N. and Tsutakawa, R. K., editors, *Statistical Numerical Integration, Contemporary Mathematics*, volume 115, pages 165–187. American Mathematical Society, Providence, RI.

Oh, M.-S. and Berger, J. O. (1993). Integration of multimodal functions by Monte Carlo importance sampling. *Journal of the American Statistical Association*, 88(422):450–456.

Owen, A. B. (1992a). A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society, Series B*, 54:541–551.

Owen, A. B. (1992b). Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452.

Owen, A. B. (1994). Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. *Annals of Statistics*, 22:930–945.

Owen, A. B. (1997). Monte Carlo variance of scrambled net quadrature. *SIAM Journal of Numerical Analysis*, 34(5):1884–1910.

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

Owen, A. B. and Zhou, Y. (1999). Adaptive importance sampling by mixtures of products of beta distributions. Technical report, Stanford University.

Patterson, H. D. (1954). The errors of lattice sampling. *Journal of the Royal Statistical Society, Series B*, 16:140–149.

Pennanen, T. and Koivu, M. (2006). An adaptive importance sampling technique. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 443–455. Springer.

Press, W. H. and Farrar, G. R. (1990). Recursive stratified sampling for multidimensional Monte Carlo integration. *Computers in Physics*, 4(2):190–195.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes: the art of scientific computing.* Cambridge University Press, Cambridge, 3rd edition.

Qin, J. (1993). Empirical likelihood in biased sample problems. *The Annals of Statistics*, 21:1182–1196.

Raghavarao, D. (1971). *Constructions and combinatorial problems in design of experiments*. Wiley, New York.

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112.

Rubinstein, R. Y. and Kroese, D. P. (2004). *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer, New York.

Schürer, R. (2004). Adaptive quasi-monte carlo integration based on MISER and VEGAS. In Niederreiter, H., editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 393–406. Springer.

Shirley, P. S. (1991). *Physically based lighting calculations for computer graphics*. PhD thesis, University of Illinois at Urbana-Champaign.

Shirts, M., Bair, E., Hooker, G., and Pande, V. (2003). Equilibrium free energies from nonequilibrium estimates using maximum likelihood methods. *Physical Letters Review*, 91(14):140601–(1–4).

Sinharay, S. and Stern, H. S. (2005). An empirical comparison of methods for computing Bayes factors in generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 14(2):415–435.

Sloan, I. H. and Joe, S. (1994). *Lattice Methods for Multiple Integration*. Oxford Science Publications, Oxford.

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–51.

Tan, Z. (2004). On a likelihood approach for Monte Carlo integration. *Journal of the American Statistical Association*, 99(468):1027–1036.

Tang, B. (1993). Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association*, 88:1392–1397.

Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in Monte Carlo free energy calculations: umbrella sampling. *Journal of computational physics*, 23(2):187–199.

Vardi, Y. (1985). Empirical distributions in selection bias models. *The Annals of Statistics*, 13:178–203.

Voter, A. F. (1985). A Monte Carlo method for determining free-energy differences and transition state theory rate constants. *Journal of Chemical Physics*, 82(4):1890–1899.

Wang, L. and Meng, X.-L. (2013). Warp bridge sampling: the next generation. Technical report, Harvard University.

Zhou, Y. (1998). *Adaptive importance sampling for integration*. PhD thesis, Stanford University.