Chapter 5

Variance reduction

The error in a direct Monte Carlo simulation goes as $\sigma/\sqrt{n}$. So there are two ways we can reduce the error. Run the simulation for a longer time, i.e., increase n or find a different formulation of the Monte Carlo that has a smaller σ. Methods that do the latter are know as variance reduction.

5.1 Antithetic variables

If X and Y are independent, then $var(X+Y) = var(X) + var(Y)$. If they are not independent the covariance enters. Letting μ_X, μ_Y denote the means of X and Y, we have

$$\begin{aligned} var(X+Y) &= E[(X+Y)^2] - (\mu_X + \mu_Y)^2 && (5.1)\\ &= E[X^2] - \mu_X^2 + E[Y^2] - \mu_Y^2 + 2(E[XY] - \mu_X\mu_Y) && (5.2)\\ &= var(X) + var(Y) + 2cov(X,Y) && (5.3) \end{aligned}$$

The covariance is $cov(X,Y) = \rho_{X,Y}\sigma_X\sigma_Y$ and ρ lies between -1 and 1. If ρ is negative the variance of $X+Y$ is smaller than the sum of their variances. Antithetic variables take advantage of this fact.

Definition 1 *Random variables X, Y on the same probability space are antithetic if they have the same distribution and their covariance is negative.*

Suppose we want to compute $\mu = E[X]$ and we can find another random variable Y such that X, Y is an antithetic pair. So $E[Y]$ is also equal to μ. Our Monte Carlo algorithm is as

follows. We generate n indendent samples $\omega_1, \cdots, \omega_n$ from the probability space and let $X_i = X(\omega_i)$ and $Y_i = Y(\omega_i)$. Our estimator for μ is then

$$\hat{\mu}_n = \frac{1}{2n} \sum_{i=1}^{n} (X_i + Y_i) \tag{5.4}$$

Obvious the mean of $\hat{\mu}_n$ is μ, so this is an unbiased estimator. Let ρ denote the correlation of X and Y. Since they have the same distribution, they have the same variance. We denote it by σ^2. So the variance of our estimator is

$$\begin{aligned} var(\hat{\mu}_n) &= \frac{1}{(2n)^2} n \; var(X_1 + Y_2) && (5.5) \\ &= \frac{1}{2n} \sigma^2 (1 + \rho) && (5.6) \end{aligned}$$

To compare this with direct Monte Carlo just using X we have to pay attention to the times involved. Recall that the relevant quantity for the quality of our MC is $\sigma^2 \tau$, where τ is the time to generate a sample.

For direct MC with just X we have to generate an ω and then evaluate X on it. For our antithetic MC we have to generate an ω and then evaluate both X and Y on it. Let τ_ω be the time required to generate an ω. We assume it takes the same time to evalute Y that it does to evaluate Y. Call that time τ_e. (e for evaluation.) Then the original MC takes time $\tau_\omega + \tau_e$, while the antithetic MC takes time $\tau_\omega + 2\tau_e$. So we need to compare $\sigma^2(\tau_\omega + \tau_e)$ for the original MC with $\sigma^2 \frac{1}{2}(1 + \rho)(\tau_\omega + 2\tau_e)$ So the antithetic is better if

$$\frac{1}{2}(1 + \rho)(\tau_\omega + 2\tau_e) < \tau_\omega + \tau_e \tag{5.7}$$

If τ_ω is neglible compared to τ_e then this simplifies to $\rho < 0$. On the other hand, if τ_e is neglible compared to τ_ω then this simplifies to $\rho < 1$ which is always true unless $Y = X$. Note that the advantage of the antithetic MC will be large only if ρ is close to -1.

If we want to find a confindence interval for our estimate, we need the variance of the antithetic estimator. We could use the calculations above. But this requires estimating ρ. We can avoid this by the following approach. Let $Z = (X + Y)/2$, and let $Z_i = (X_i + Y_i)/2$. We can think of our antithetic Monte Carlo as just generating n samples of Z. Then we compute the sample variance of the sample $Z_1, \cdots, Z_n$ and just do a straightforward confidence interval.

Of course this is only useful if we can find antithetic pairs. We start with a trivial example. We want to compute

$$\mu = \int_0^1 f(x) dx = E[f(U)] \tag{5.8}$$

where U is a uniform random variable on $[0,1]$. So we are computing the mean of $X = f(U)$. Suppose f is an increasing function. Then it might be helpful to balance a value of U in $[0,1/2]$ with its "reflection" $1-U$. So take $Y = F(1-U)$. This has the same distribution (and hence the same mean) as X since $1-U$ is uniform on $[0,1]$. A fancy way to say this is that the uniform probability measure on $[0,1]$ is invariant under the map $1 \to 1-x$ on $[0,1]$.

So we consider the following general set-up. We assume there is a map $R : \Omega \to \Omega$ under which the probability measure is invariant, i.e., $P = P \circ R$. We define $Y(\omega) = X(R\omega)$. Then Y has the same distribution as X and hence the same mean. We need to study the correlation of X and Y to see if this will be useful. Define

$$X_e(\omega) = \frac{1}{2}[X(\omega) + X(R\omega)], \tag{5.9}$$

$$X_o(\omega) = \frac{1}{2}[X(\omega) - X(R\omega)] \tag{5.10}$$

Then $X = X_o + X_e$, and we can think of this as a decomposition of X into its even and odd parts with respect to R. (Note $X_e(R\omega) = X_e(\omega)$, $X_o(R\omega) = -X_o(\omega)$.) The invariance of P under R implies that

$$E[X_e] = \mu, \quad E[X_o] = 0, \quad E[X_e X_o] = 0 \tag{5.11}$$

Thus X_e and X_o are uncorrelated. This is weaker than being independent, but it does imply the variance of their sum is the sum of their variances. So if we let σ_e^2 and σ_o^2 be the variances of X_e and X_o, then $\sigma^2 = \sigma_e^2 + \sigma_o^2$, where σ is the variance of X. Note that the variance of Y is also equal to σ^2. A little calculation shows that ρ, the correlation between X and Y, is given by

$$\rho = \frac{\sigma_e^2 - \sigma_o^2}{\sigma_e^2 + \sigma_o^2} \tag{5.12}$$

Thus Y will be a good antithetic variable if σ_e is small compared to σ_o. This will happen if X is close to being an odd function with respect to R.

Literature seems to say that if $X = f(U)$ where U is a vector of i.i.d. uniform on $[0,1]$ and f is increasing then $Y = f(1-U)$ is a good antithetic RV where in $1-U$, 1 means the vector with 1's in all the components.

Stop - Wed, 2/3

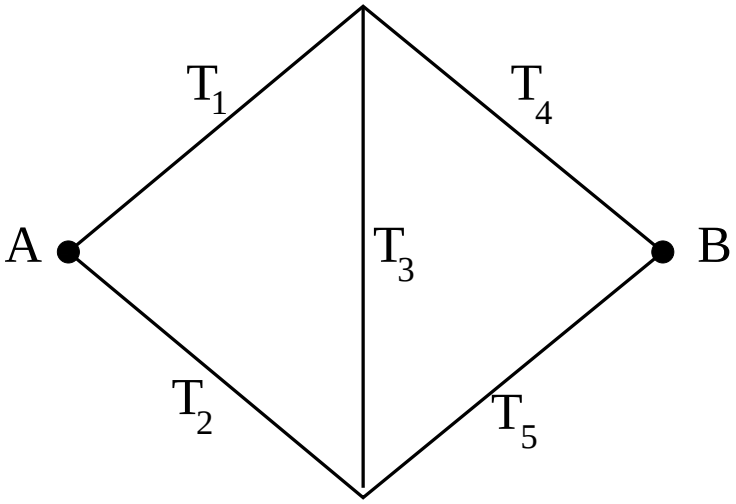

Figure 5.1: Network example. We seek the quickest path from A to B.

Network example from Kroese The times T_i are independent and uniformly distributed but with different ranges:
T_1 uniform on $[0, 1]$
T_2 uniform on $[0, 2]$
T_3 uniform on $[0, 3]$
T_4 uniform on $[0, 1]$
T_5 uniform on $[0, 2]$
The network is small enough that you can find the mean time of the quickest path analytically. It is

$$\mu = \frac{1339}{1440} \approx 0.92986 \tag{5.13}$$

Let U_1, U_2, U_3, U_4, U_5 be independent, uniform on $[0, 1]$. Then we can let $T_1 = U_1, T_2 = 2 * U_2$, etc. And the quickest time can be written as a function $X = h(U_1, U_2, U_3, U_4, U_5)$. We let $Y = h(1 - U_1, 1 - U_2, 1 - U_3, 1 - U_4, 1 - U_5)$, and then $Z = (X + Y)/2$. Our antithetic estimator is

$$\frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(X_i + Y_i) \tag{5.14}$$

where X_i and Y_i use the same U vector. Some simulation shows that without using the antithetic pair, the variance of X is approximately 0.158. Another simulation using the antithetic pair shows that the variance of Z is approximately 0.0183. The error is proportional to the square root of the variance, so the error is reduced by a factor of $\sqrt{0.158/0.0183}$. But we must keep in mind that it takes twice as long to generate a sample of

Z as it does a sample of X. So for a fixed amount of CPU time we will have half as many samples of Z which means a factor of $\sqrt{2}$ for the error. So the true reduction in the error is a factor of $\sqrt{0.158/(2 * 0.0183)}$ which is just over 2. But keep in mind that to reduce the error by a factor of 2 by increasing the number of samples would require generating 4 times as many samples. So if we think in terms of how long it takes to reach a given error level, then the antithetic method has reduced the computation time by a factor of 4.

5.2 Control variates

Consider the bridge example again. The times T_1 and T_4 are both uniformly distributed on $[0, 1]$ while the other three times are uniformly distributed on larger intervals. So we expect that the quickest path will be the path through bonds 1 and 4 with fairly high probability. In other words, if we let X be the minimum time for the full network and let $Y = T_1 + T_4$, then Y will be equal to X with high probability. Note that we know the mean of Y. Can we take advantage of this to improve our Monte Carlo method? Let ν be the known mean of Y. (Of course, $\nu = 1/2 + 1/2 = 1$.) Then $\mu = E[X - (Y - \nu)]$. So we can do a Monte Carlo simulation of $X - (Y - \nu)$ rather than X. The hope is that $X - (Y - \nu)$ has a smaller variance since it equals ν most of the time.

The general setup is as follows. We want to compute $\mu = E[X]$. We have another random variable Y on the same probability space and we know its mean. Call it $\nu = E[Y]$. (Note that μ is unknown, ν is known.) We generate a random sample $\omega_1, \omega_2, \cdots, \omega_n$ and evaluate X and Y on them. So let $X_i = X(\omega_i)$ and $Y_i = Y(\omega_i)$. Now define

$$\hat{l}_n = \frac{1}{n} \sum_{i=1}^{n} [X_i - \alpha(Y_i - \nu)] \tag{5.15}$$

where α is a parameter. In our discussion of the brigde example we took $\alpha = 1$, but now we allow a more general choice of the new estimator. Note that $E[\hat{l}_n] = \mu$, i.e., for any choice of α this is an unbiased estimator of μ.

Let ρ denote the correlation of X and Y. Let σ_X^2 and σ_Y^2 be the variances of X and Y. The variance of our estimator is

$$var(\hat{l}_n) = \frac{1}{n} var(X - \alpha Y) \tag{5.16}$$

We have

$$var(X - \alpha Y) = \sigma_X^2 + \alpha^2 \sigma_Y^2 - 2\alpha\rho\sigma_X\sigma_Y \tag{5.17}$$

We can use any α we want, so we choose α to minimize this variance. The minimizing α is given by

$$\alpha_0 = \frac{\rho\sigma_X}{\sigma_Y} = \frac{cov(X,Y)}{\sigma_Y^2} \tag{5.18}$$

in which case

$$var(X - \alpha_0 Y) = \sigma_X^2(1-\rho^2) \tag{5.19}$$

So the variance of our estimator is $\sigma_X^2(1-\rho^2)/n$. So we have reduced the variance by a factor of $1-\rho^2$. So the method works well if ρ is close to 1 or -1.

Note that to compute the optimal α we need to know σ_X, σ_Y, and ρ. We might know σ_Y, but we almost certainly will not know σ_X or ρ. So we have to use our sample to estimate them. We estimate σ_X^2 (and σ_Y^2 if needed) with the usual sample variance. And we estimate ρ with

$$\hat{\rho}_n = \frac{1}{n}\sum_{i=1}^{n}[X_iY_i - \overline{X}_n\nu] \tag{5.20}$$

where $\overline{X}_n$ is the samples mean of X.

In the above we have assumed that the mean of Y is known exactly. Even if we do not know it exactly, but just have a good approximation we can still use the above. If it is much faster to compute the control RV Y than the original RV X, then we could use a preliminary Monte Carlo to compute a good approximation to the mean of Y and then do the above Monte Carlo to get the mean of X.

We look at the last paragraph in more detail and think of what we are doing as following. We want to compute $E[X]$. We have another random variable Y such that we think the varince of $X-Y$ is small. We write X as $(X-Y)+Y$ and try to compute the mean of X by computing the means of $X-Y$ and Y separately. Let σ_x^2, σ_Y^2 be the variances of X and Y. Let σ_{X-Y} be the variance of $X-Y$, which hopefully is small. Let τ_X and τ_Y be the time it take to generate samples of X and Y. We assume we have a fixed amount T of CPU time. If we do ordinary MC to compute $E[X]$, then we can compute T/τ_X samples and the square of the error will be $\sigma_X^2\tau_X/T$.

Now suppose we do two independent Monte Carlo simulations to compute the means of $X-Y$ and Y. For the $X-Y$ simulation we generate n_1 samples and for the Y simulation we generate n_2 samples. These numbers are constrained by $n_1(\tau_X+\tau_Y)+n_2\tau_Y = T$. We assume that τ_Y is much smaller than τ_X and replace this constraint by $n_1\tau_X + n_2\tau_Y = T$. Since our two Monte Carlos are independent, the square of the error is the sum of the squares of the errors of the two Monte Carlos, i.e.,

$$\frac{\sigma_{X-Y}^2}{n_1} + \frac{\sigma_Y^2}{n_2} \tag{5.21}$$

Now we minimize this as a function of n_1 and n_2 subject to the constraint. (Use Lagrange multiplier or just use the constraint to solve for n_2 in terms of n_1 and turn it into a one variable minimization problem.) You find that the optimal choice of n_1, n_2 is

$$n_1 = \frac{T\sigma_{X-Y}}{\sqrt{\tau_X}(\sigma_{X-Y}\sqrt{\tau_X} + \sigma_Y\sqrt{\tau_Y})}, \tag{5.22}$$

$$n_2 = \frac{T\sigma_Y}{\sqrt{\tau_Y}(\sigma_{X-Y}\sqrt{\tau_X} + \sigma_Y\sqrt{\tau_Y})} \tag{5.23}$$

which gives a squared error of

$$\frac{1}{T}(\sigma_{X-Y}\sqrt{\tau_X} + \sigma_Y\sqrt{\tau_Y})^2 \tag{5.24}$$

If σ_{X-Y} is small compared to σ_X and σ_Y and τ_Y is small compared to τ_X, then we see this is a big improvement over ordinary MC.

Network example We return to our network example. For the control variable we use $Y = T_1 + T_4$. So we do a Monte Carlo simulation of $Z = X + (Y - \nu)$ where $\nu = E[Y] = 1$. We find that the variance of Z is approximately 0.0413. As noted earlier the variance of X is approximately 0.158. So the control variate approach reduced the variance by a factor of approximately 3.8. This corresponds to a reduction in the error by a factor of $\sqrt{3.8}$. If we want a fixed error level then the use of a control variate reduces the computation time by a factor of 3.8.

Note that you do not have to know what α you want to use before you do the MC run. You can compute the sample means and sample variances for X and Y separately as well as an estimator for ρ. Then at the end of the run you can use the sample variances and estimator for ρ to compute an estimator for the best α. Note, however, that if you do this your α now depends on all the samples and so the samples $X_i - \alpha Y_i$ are not independent. So the usual method of deriving a confidence interval is not legit. If you really want to worry about this see section 4.2 of Fishman's *Monte Carlo: Concepts, Algorithms, and Applications*. I would be surprised if it matters unless n is small.

It is possible to use more than one control variable. Let $\vec{Y} = (Y^1, \cdots, Y^d)$ be vector of random variables. We assume we know their means $\nu^i = E[Y^i]$. Then our estimator is

$$\hat{l}_n = \frac{1}{n}\sum_{i=1}^{n}[X_i - (\vec{\alpha}, \vec{Y}_i - \vec{\nu})] \tag{5.25}$$

where $\vec{\alpha}$ is a vector of parameters and $(\vec{\alpha}, \vec{Y} - \vec{\nu})$ denotes the inner product of that vector and $\vec{Y} - \vec{\nu}$. To find the optimal α we need to minimize the variance of $X - (\vec{\alpha}, Y)$.

$$var(X - (\vec{\alpha}, Y)) = cov(X - (\vec{\alpha}, Y), X - (\vec{\alpha}, Y)) \tag{5.26}$$

$$= var(X, X) - 2\sum_{i=1}^{2}\alpha_i cov(X, Y_i) + \sum_{i,j=1}^{2}\alpha_i\alpha_j cov(Y_i, Y_j) \tag{5.27}$$

Let Σ be the matrix with entries $cov(Y_i, Y_j)$, i.e., the covariance matrix of the control variates. Let $\vec{C}$ be the vector of covariances of X and the Y_i. Then the above is

$$var(X,X) - 2\sum_{i=1}^{2} \alpha_i C_i + \sum_{i,j=1}^{2} \alpha_i \alpha_j \Sigma_{i,j} = var(X,X) - 2(\vec{\alpha}, \vec{C}) + (\vec{\alpha}, \Sigma\vec{\alpha}) \tag{5.28}$$

Optimal α is

$$\vec{\alpha}_0 = \Sigma^{-1}\vec{C} \tag{5.29}$$

5.3 Stratified sampling

To motivate stratified sampling consider the following simple example. We want to compute

$$I = \int_0^1 f(x)\, dx \tag{5.30}$$

On the interval $[0, 1/2]$ the function $f(x)$ is nearly constant, but on the interval $[1/2, 1]$ the function varies significantly. Suppose we wanted to use Monte Carlo to compute the two integrals

$$\int_0^{1/2} f(x)\, dx, \quad \int_{1/2}^1 f(x)\, dx \tag{5.31}$$

The Monte Carlo for the first integral will have a much smaller variance than the Monte Carlo for the second integral. So it would make more sense to spend more time on the second integral, i.e., generate more samples for the second integral. However, under the usual Monte Carlo we would randomly sample from $[0, 1]$ and so would get approximately the same number of X_i in $[0, 1/2]$ and in $[1/2, 1]$. The idea of stratified sampling is to divide the probability space into several regions and do a Monte Carlo for each region.

We now turn to the general setting. As always we let Ω be the probability space, the set of possible outcomes. We partition it into a finite number of subsets Ω_j, $j = 1, 2, \cdots, J$. So

$$\Omega = \cup_{j=1}^{J} \Omega_j, \quad \Omega_j \cap \Omega_k = \emptyset \; if \; j \neq k \tag{5.32}$$

We let $p_j = P(\Omega_j)$ and let P_j be $P(\cdot|\Omega_j)$, the probability measure P conditioned on Ω_j. We assume that the probabilities p_j are known and that we can generate samples from the conditional probability measures $P(\cdot|\Omega_j)$. The sets Ω_j are called the strata. Note that the partition theorem says that

$$P(\cdot) = \sum_{j=1}^{J} p_j P(\cdots|\Omega_j), \tag{5.33}$$

$$E[X] = \sum_{j=1}^{J} p_j E[X|\Omega_j] \tag{5.34}$$

We are trying to compute $\mu = E[X]$. We will generate samples in each strata, and the number of samples from each strata need not be the same. So let n_j, $j = 1, 2, \cdots, J$ be the number of samples from strata j. Let $X_i^j, i = 1, 2, \cdots, n_j$ be the samples from the jth strata. Then our estimator for μ is

$$\hat{\mu} = \sum_{j=1}^{J} \frac{p_j}{n_j} \sum_{i=1}^{n_j} X_i^j \tag{5.35}$$

Note that the expected value of X_i^j is $E[X|\Omega_j]$. So the mean of $\hat{\mu}$ is

$$E[\hat{\mu}] = \sum_{j=1}^{J} p_j E[X|\Omega_j] = E[X] \tag{5.36}$$

where the last equality follows from the partition theorem.

Let

$$\mu_j = E[X|\Omega_j], \quad \sigma_j^2 = E[X^2|\Omega_j] - \mu_j^2 \tag{5.37}$$

The quantity σ_j^2 is often denoted $var(X|\Omega_j)$. It is the variance of X if we use $P(\cdot|\Omega_j)$ as the probability measure instead of $P(\cdot)$.

We write out our estimator as

$$\hat{\mu} = \sum_{j=1}^{J} p_j \hat{\mu}_j, \quad \hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} X_i^j \tag{5.38}$$

We assume that we generate sample from different strata in an independent fashion. So the random variables $\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_J$ are independent. The variance of $\hat{\mu}_j$ is σ_j^2. So we have

$$var(\hat{\mu}) = \sum_{j=1}^{J} p_j^2 \frac{\sigma_j^2}{n_j} \tag{5.39}$$

Stop - Mon, 2/8

How do we choose the n_j? One possible choice is *proportional allocation*. Letting N denote the total number of samples we will generate, we take $n_j = p_j N$. (Of course, we have to round this to the nearest integer.) This gives

$$var(\hat{\mu}) = \frac{1}{N} \sum_{j=1}^{J} p_j \sigma_j^2 \tag{5.40}$$

To compare this with the variance of ordinary MC, we do a little computation.

$$\begin{aligned}
\sum_{j=1}^{J} p_j \sigma_j^2 &= \sum_{j=1}^{J} p_j E[(X-\mu_j)^2|\Omega_j] && (5.41)\\
&= \sum_{j=1}^{J} p_j E[((X-\mu)+(\mu-\mu_j))^2|\Omega_j] && (5.42)\\
&= \sum_{j=1}^{J} p_j \left[E[((X-\mu)^2|\Omega_j] + (\mu-\mu_j))^2 + 2(\mu-\mu_j)E[X-\mu|\Omega_j\right] && (5.43)
\end{aligned}$$

Note that $\sigma^2 = \sum_j p_j E[(X-\mu)^2|\Omega_j]$, and $E[X-\mu|\Omega_j] = \mu_j - \mu$. So the above reduces to

$$\sum_{j=1}^{J} p_j \sigma_j^2 = \sigma^2 - \sum_{j=1}^{J} p_j (\mu-\mu_j)^2 \tag{5.44}$$

So using proportional allocation the variance of the Monte Carlo using strata is smaller than the variance of plain Monte Carlo unless the μ_j are all equal to μ We also see that we should try to choose the strata so that the means within the strata are far from the overall mean.

But is proportional allocation optimal? Recall our motivating example. We probably should sample more in the Ω_j with higher variance. We can find the optimal choice of n_j. We want to minimize $var(\hat{\mu})$ subject to the constraint that the total number of samples is fixed, i.e.,

$$\sum_{j=1}^{J} n_j = N \tag{5.45}$$

This is a straightforward Lagrange multiplier problem. Let

$$\begin{aligned}
f(n_1, n_2, \cdots, n_J) &= \sum_{j=1}^{J} p_j^2 \frac{\sigma_j^2}{n_j}, && (5.46)\\
g(n_1, n_2, \cdots, n_J) &= \sum_{j=1}^{J} n_j && (5.47)
\end{aligned}$$

We want to minimize f subject to $g = N$. The minimizer will satify

$$\nabla f(n_1, n_2, \cdots, n_J) = -\lambda \nabla g(n_1, n_2, \cdots, n_J) \tag{5.48}$$

for some λ. So for $j = 1, 2, \cdots, J$

$$-p_j^2 \frac{\sigma_j^2}{n_j^2} = -\lambda \tag{5.49}$$

Solving for n_j,

$$n_j = \frac{1}{\sqrt{\lambda}} p_j \sigma_j \tag{5.50}$$

Thus

$$n_j = c p_j \sigma_j \tag{5.51}$$

where the constant c is chosen to make the sum of the n_j sum to N. So $c = N/\sum_j p_j \sigma_j$. This choice of n_j makes the variance of the estimate

$$var(\hat{\mu}) = \frac{1}{N} [\sum_{j=1}^{J} p_j \sigma_j]^2 \tag{5.52}$$

Note that the Cauchy Schwarz inequality implies this is less than or equal to the variance we get using proportional allocation and it is equal only if the σ_j are all the same. Of course, to implement this optimal choice we need to know all the σ_j whereas the proportional allocation does not depend on the σ_j.

Suppose the time to generate a sample depends on the strata. Let τ_j be the time to generate a sample in the jth strata. Then if we fixed the amount of computation time to be T, we have $\sum_j n_j \tau_j = T$. Then using a Lagrange multiplier to find the optimal n_j we find that n_j should be proportional to $p_j \sigma_j / \sqrt{\tau_j}$.

Example: rainfall example from Owen.

Example: Network example. Partition each uniform time into 4 intervals, so we get 4^5 strata.

5.4 Conditioning

To motivate this method we first review a bit of probability. Let X be a random variable, $\vec{Z}$ a random vector. We assume X has finite variance. We let $E[X|\vec{Z}]$ denote the conditional expectation of X where the conditioning is on the σ-field generated by $\vec{Z}$. Note that $E[X|\vec{Z}]$ is a random variable. We assume denote its variance by $var(E[X|\vec{Z}])$.

We define the conditional variance of X to be

$$var[X|\vec{Z}] = E[X^2|\vec{Z}] - (E[X|\vec{Z}])^2 \tag{5.53}$$

Note that $var[X|\vec{Z}]$ is a random variable. There is a conditional Cauchy Schwarz inequality which says that this random variable is always non-negative. A simple calculation gives

$$var(X) = E[var[X|\vec{Z}]] + var(E[X|\vec{Z}]) \tag{5.54}$$

In particular this shows that $E[X|\vec{Z}]$ has smaller variance than X.

The conditional expectation $E[X|\vec{Z}]$ is a function of $\vec{Z}$. There is a Borel-measurable function $h : R^d \to R$ such that $X = h(\vec{Z})$. Now suppose that it is possible to explicitly compute $E[X|\vec{Z}]$, i.e., we can explicitly find the function h. Suppose also that we can generate samples of $\vec{Z}$. One of the properties of conditional expectation is that the expected value of the conditional expectation is just the expected value of X. So we have

$$\mu = E[X] = E[E[X|\vec{Z}]] = E[h(\vec{Z})] \tag{5.55}$$

So we have the following Monte Carlo algorithm. Generate samples $\vec{Z}_1, \cdots, \vec{Z}_n$ of $\vec{Z}$. Compute $h(\vec{Z}_1), \cdots, h(\vec{Z}_n)$. Then the estimator is

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} h(\vec{Z}_i) \tag{5.56}$$

Of course the non-trivial thing is to find a random vector $\vec{Z}$ for which we can explicitly compute $E[X|\vec{Z}]$.

Example: Let $X_1, \cdots, X_d$ be independent random variables with exponential distributions and $E[X_i] = 1/\lambda_i$. We want to compute the probability that the largest of the d random variables is X_1, i.e., we want to compute

$$\mu = P(X_i < X_1, i = 2, \cdots, d) \tag{5.57}$$

We are particularily interested in the case that λ_1 is large compared to the other λ_i. In this case X_1 is usually small compared to the other X_i, so so the the probability μ will be tiny and very hard to compute accurately with an ordinary MC.

Suppose we condition on X_1. Then keeping in mind that the X_i are independent, we have

$$\begin{aligned} P(X_i < X_1, i = 2, \cdots, d|X_1 = x_1) &= P(X_i < x_1, i = 2, \cdots, d|X_1 = x_1) && (5.58)\\ &= P(X_i < x_1, i = 2, \cdots, d) && (5.59)\\ &= \prod_{i=2}^{d} P(X_i < x_1) && (5.60) \end{aligned}$$

Note that if X has an exponential distribution with mean λ, then $P(X < x) = 1 - e^{-\lambda x}$. Define $F(x) = 1 - e^{-x}$. So $P(X < x) = F(\lambda x)$. Then the above becomes

$$P(X_i < X_1, i = 2, \cdots, d | X_1 = x_1) = \prod_{i=2}^{d} F(\lambda_i x_1) \tag{5.61}$$

Now the probability we want is

$$\mu = P(X_i < X_1, i = 2, \cdots, d) = E[P(X_i < X_1, i = 2, \cdots, d | X_1)] = E[\prod_{i=2}^{d} F(\lambda_i X_1)] \tag{5.62}$$

If λ_1 is large compared to the other λ_i, then $\lambda_i X_1$ will typically be small and so the random variable in the expectation is very small. But that is ok. It is much better that trying to do MC on an indicator function that is 0 with very high probability.

Example In the network example, take $\vec{Z} = (T_1, T_2, T_3)$. It is possible, but not trivial, to compute $E[X|T_1, T_2, T_3]$ where X is the minimum time to get from A to B.

Stop - Wed, 2/10
